Isomorphism of Polynomials : New Results

Charles Bouillaguet[1], Jean-Charles Faugère[2,3],
Pierre-Alain Fouque[1] and Ludovic Perret[3,2]

[1] Ecole Normale Supérieure
{charles.bouillaguet, pierre-alain.fouque}@ens.fr
[2] INRIA Salsa
jean-charles.faugere@inria.fr
[3] Lip 6
ludovic.perret@lip6.fr

Abstract. In this paper, we investigate the difficulty of the Isomorphism of Polynomials (IP) Problem as well as one of its variant IP1S. The Isomorphism of Polynomials is a well-known problem studied in multivariate cryptography. It is related to the hardness of the key recovery of some cryptosystems. The problem is the following: given two families of multivariate polynomials $\mathbf{a}$ and $\mathbf{b}$, find two invertible linear (or affine) mappings S and T such that $\mathbf{b} = T \circ \mathbf{a} \circ S$. For IP1S, we suppose that T is the identity. It is known that the difficulty of such problems depends on the structure of the polynomials (*i.e.*, homogeneous, or not) and the nature of the transformations (affine, or linear). Here, we analyze the different cases and propose improved algorithms. We precisely describe the situation in term of complexity and sufficient conditions so that the algorithms work. The algorithms presented here combine linear algebra techniques, including the use of differentials, together with Gröbner bases. We show that random instances of IP1S with quadratic polynomials can be broken in time $\mathcal{O}\left(n^6\right)$, where n is the number of variables, independently of the number of polynomials. For IP1S with cubic polynomials, as well as for IP, we propose new algorithms of complexity $\mathcal{O}\left(n^6\right)$ if the polynomials of $\mathbf{a}$ are inhomogeneous and S, T linear. In all the other cases, we propose an algorithm that requires $\mathcal{O}\left(n^6 q^n\right)$ computation. Finally, if $\mathbf{a}$ and $\mathbf{b}$ have a small number of non-trivial zeros, the complexity solving the IP instance is reduced to $\mathcal{O}\left(n^6 + q^n\right)$. This allows to break a public-key authentication scheme based on IP1S, and to break all the IP challenges proposed by Patarin in 1996 in practical time: the more secure parameters require less than 6 months of computations on 10 inexpensive GPUs. A consequence of our results is that HFE can be broken in polynomial time if the secret transforms S and T are linear and if the internal polynomial is made public and contains linear and constant terms.

1 Introduction

Multivariate cryptography is concerned with the use of multivariate polynomials over finite fields in cryptographic schemes. The use of polynomial systems in cryptography dates back to the mid eighties with the design of C* [34], and many other proposals appeared afterwards [38,39,40,30,45]. The security of multivariate schemes is in general related to the difficulty of solving random or structured systems of multivariate polynomials. This problem has been proved to be NP-complete [25], and it is conjectured [2] that systems of random polynomials are hard to solve in practice. As usual when a trapdoor must be embedded in a hard problem, easy instances are transformed into random-looking instances using secret transformations. In multivariate cryptography, it is common to turn an easily-invertible collection of polynomials $\mathbf{a}$ into an apparently random one $\mathbf{b}$. It is then assumed that, being supposedly indistinguishable from random, $\mathbf{b}$ should be hard to solve. The structure-hiding transformation is very often the composition with linear (or affine) invertible mappings S and T: $\mathbf{b} = T \circ \mathbf{a} \circ S$. The matrices S and T are generally part of the secret-key.

The Isomorphism of Polynomials (IP) is the problem of recovering the secret transformations S and T given $\mathbf{a}$ and $\mathbf{b}$. It is a fundamental problem of multivariate cryptography, since its hardness implies the hardness of the key-recovery for various multivariate cryptosystems. With the notable exception of the HFE encryption and signature scheme [36], most of the multivariate schemes that were broken fell victim of a key-recovery attack, *i.e.*, of a *customized* IP algorithm. Notorious examples include C* [34], the traitor tracing scheme proposed by Billet and Gilbert [7], the SFLASH signature scheme [39], the square-vinegar signature scheme [1] and the Square encryption scheme [11]. The hardness of the key-recovery problem for HFE does not rely on the hardness of IP, since the easily-invertible mapping $\mathbf{a}$ is kept secret. One could however consider a variant of HFE where $\mathbf{a}$ is made public; its security would rely directly on the hardness of IP. In this paper, we do not target a specific cryptosystem nor use the fact that the system $\mathbf{a}$ is built in a special way so that it is efficient to invert. We in fact consider both $\mathbf{a}$ and $\mathbf{b}$ to be random collections of polynomials, which is to some extent a worst-case assumption.

An important special case of IP is the *IP problem with one secret* (IP1S for short). Patarin suggested in 1996 [36] to construct a zero-knowledge public-key authentication scheme relying on the hardness of IP1S, inspired by the Zero-Knowledge proof system for Graph Isomorphism of [28]. The proposed parameters lead to small key sizes (for instance to secret and public keys of 256 bits each), as the complexity of the problem was believed to be exponential. These parameters have not been broken so far. The IP1S problem is also interesting from a complexity-theoretic point of view: it has been proved [41] that IP1S is *Graph Isomorphism-hard* (GI-hard for short). This leads Patarin *et al.* to claim that IP1S (and *a fortiori* IP) is unlikely to be solvable in polynomial time, because no polynomial algorithm is known for GI in spite of more than forty years of research. On the other hand, GI is not known to be NP-complete. Forging hard instances of the GI problem is pretty non-trivial, and there are powerful heuristics as well as expected linear time algorithms for random graphs [21]. This compromises the use of GI as an authentication mechanism, and was part of the motivation for introducing IP1S as an alternative.

Related Work. As already explained, the IP problem has been introduced by Patarin in [36,37]. In this section, we summarize existing results on the IP and IP1S problems.

The first algorithm for IP, known as the “To and Fro” technique, is due to Courtois *et al.* [41]. In its primitive form, this algorithm assumes the ability to inverse the polynomial systems, and has therefore an exponential complexity. Moreover, the image of S has to be known on at least one point, and is found via exhaustive search if no other solution applies. An improved, birthday-based version of this algorithm is claimed to find a few relations on S in time and space $\mathcal{O}\left(q^{n/2}\right)$.

In [20], Perret and Faugère present a new technique for solving IP when S and T are linear (as opposed to affine) invertible mappings. The idea is to model the problem as an algebraic system of equations and solve it by means of Gröbner bases [9,12]. In practice, the technique turns out to be efficient for instances of IP where the coefficients of all the monomials of all degree are randomly chosen in $\mathbb{F}_q$. For random instances of IP, the practical complexity of [20] is estimated to be $\mathcal{O}\left(n^9\right)$. This complexity corresponds to the empirical observation that the maximum degree reached during the Gröbner basis computation [17,18] seems to bounded from above by 3. Note this result no longer holds as soon as the IP instances are “structured”. Typically, [20] observed that *homogeneous* instances of IP (in which the polynomials of $\mathbf{a}$ and $\mathbf{b}$ are homogeneous) are much harder to solve in practice.

To conclude about the full IP problem, we mention the work of Fouque, Macario-Rat and Stern in [22]. They proposed a dedicated algorithm to solve the IP problem arising in C* [34] (and variants). Using a differential technique, they mounted an efficient polynomial-time key-recovery against C* [34] and SFLASH [39]. In this paper, we will combine ideas from this work, namely the use of differentials, together with Gröbner bases techniques of [20].

IP1S is a subcase of IP, thus most of the algorithms presented above could be applied to the one secret variant almost directly. However, several algorithms were developed to exploit the specificities of IP1S. It was shown in [41] that linear equations on the coefficients of the secret matrix can be obtained in some cases. Our quadratic IP1S technique is based on an extension of this observation. To our knowledge, the first algorithm dedicated to IP1S can be found in [26]. We briefly explain the idea in the linear case. The authors of [26] remarked that each row of a matrix solution of IP1S verifies an algebraic system of equations. They then used an exhaustive search to find the solutions of such system. Soon after, this technique has been improved by [13] who replaced this exhaustive search by a Gröbner basis computation. The improvement yielded by this replacement is as significant as the gain obtained when comparing Gröbner basis and exhaustive search for solving random algebraic systems. It is negligible over small field (*i.e.*, typically, $\mathbb{F}_2$), but significant for instances of IP1S over large fields. However, the complexity of those algorithms remains exponential by nature.

Finally, [42] shows that the affine and linear variants of IP1S are equivalent, *i.e.*, one can without loss of generality restrict our attention to the linear case. In addition, a new approach for solving IP1S using the Jacobian matrix was proposed. The algorithm is polynomial when the number u of polynomials in $\mathbf{a}$ and $\mathbf{b}$ is equal to the number of variables n. However, when $u < n$, the complexity of this approach is not well understood. Moreover, when the number of polynomials is very small, for instance $u = 2$, this algorithm is totally inefficient. Whilst the case $u = n$ is the classical setting for IP, it is less interesting for IP1S. Indeed, the main application of IP1S is the authentication scheme proposed in [41]. The public key being composed of two sets of u polynomials, it is interesting to keep the number of polynomials as small as possible (1 or

2). For such parameters, the authentication mechanism based on IP1S looks appealing in terms of key size. In this paper, we will break all the proposed parameters. As a consequence, we believe that this makes the IP1S authentication scheme much less interesting for practical purposes.

All in all, the existing literature on the IP problem discussed above can be split in two categories: *heuristic* algorithms with "known" complexity and unknown success probability [41], and *rigourous* algorithms that always succeeds but with unknown complexity [20,42,13,26]. This situation makes it very difficult, if not plainly impossible to compare these algorithms based on their theoretical features. The class of instances that can be solved by a given algorithm of the first type is in general not known. Conversely, the class of instances over which an algorithm of the second type terminates quickly are often not known as well. This lead the authors of IP algorithms to measure the efficiency of their techniques in practice, or even not to measure it at all [41]. Several IP and IP1S challenges were proposed by Patarin in [36], and can be used to measure the progress accomplished since their introduction. Also, the lack of theoretical understanding of the previous techniques makes it quite difficult to get a clear picture of the secure and broken ranges of parameters.

Our Results. In this paper, we present four new algorithms: the first one deals with quadratic IP1S instances, the second with cubic IP1S instances, and the last two ones with linear inhomogeneous IP instances. These algorithms combine the two weapons that have dealt a severe blow to multivariate cryptography: linear algebra and Gröbner bases.

On the practical side, these algorithms are efficient: random quadratic IP1S instances, random inhomogeneous linear cubic IP1S instances and random inhomogeneous linear quadratic IP instances can all be broken in practical time for any size of the parameters. In particular, all the quadratic IP1S challenges are now broken in a few seconds. The biggest cubic IP1S challenge is broken in 2 CPU-month. The IP1S authentication scheme is thus broken beyond repair in the quadratic case. In the case of cubic IP1S, the security parameter have to be seriously reconsidered, which makes the scheme much less attractive in terms of key size. HFE with linear secret transformations S and T, and public internal polynomial is completely insecure.

A rigorous analysis of our algorithms is both necessary and tricky. When generating linear equations, special care has to be taken to count how many of them are independent. The recent history of algebraic cryptanalysis taught us that failure to do so may have drastic consequences. On the other hand, the complexity of Gröbner bases computation, even though a bit more well-understood now in the generic case, is still often a delicate matter. Most algorithms presented in this paper belongs to "rigorous" type (with the exception of the heuristic for linear inhomogeneous IP presented in section 4.1).

A distinctive feature of our algorithms compared to the previous state of affairs, and one of our main theoretical contribution, is that we characterize the class of instances that can be solved by each algorithm in polynomial time. We then estimate as rigorously as possible the probability that a random instance falls in this class. The rigorous analysis of the algorithm of section 4.2 shows that it solves random linear inhomogeneous instances of IP in time $\mathcal{O}\left(n^6\right)$ with non-negligible probability, and gives *en route* a theoretical justification to the empirical observation of a polynomial behavior for the same subproblem in [20].

On the theoretical side again, we clarify the question of the relative complexity of various classes of IP subproblems. We identify the parameters that make an IP instance easy or hard. Such parameters include the homogeneousness of the polynomials, their degree, the parity of the characteristic of the field, and the presence of an affine part in the linear masks S and T. We extend the affine/linear equivalence result of [42] to the full IP problem. A consequence is that the hardest IP instances are not the affine ones, but the linear homogeneous ones. All the previous algorithms that were dealing with the affine case [41,26] first guessed the affine component of S or T in order to reduce the affine instance to a linear one; these algorithms were exponential by nature. Our algorithm for random quadratic IP1S instances is polynomial even in the affine case. We discuss various technique to deal with the linear homogeneous IP case, but our techniques are all exponential.

We lastly revisit an algorithm published twelve years ago in [41]. The algorithm is birthday-based and supposedly has an interesting complexity of $\mathcal{O}\left(q^{n/2}\right)$. It has never been implemented, and we show that several heuristic assumptions used to establish its complexity and success probability are in fact not true. As a consequence, we show that its domain of applicability is restricted to $\mathbb{F}_2$, and that its complexity is higher than expected. This advocates for a more in-depth analysis.

Organisation of the paper. In section 2, we present some basic ingredients that explain the Faugère and Perret algorithm and our basic strategy to solve the IP problem. Then, in section 3, we present our results concerning the IP1S problem with quadratic and cubic polynomials. Finally, in section 4, we present and analyze our algorithms to solve the IP problem with two secrets.

2 Preliminaries

In this section, we remind the definition of the differential of a function and we present a basic fact which is the core of the Perret-Faugère algorithm [20]. Then, we present our meta-algorithm for the IP problem.

Differentials. We denote by $\mathrm{D}f : Dom^2 \to Rng$ the *differential* of a function $f : Dom \to Rng$. $\mathrm{D}f$ is defined by:

$$\mathrm{D}f(\mathbf{x}, \mathbf{y}) = f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}) - f(\mathbf{y}) + f(0)$$

It is easy to see that $\mathrm{D}f(\mathbf{x}, \mathbf{y}) = \mathrm{D}f(\mathbf{y}, \mathbf{x})$. If f is a polynomial of total degree d, then $\mathrm{D}f$ is a polynomial of total degree d, but of degree $d-1$ in $\mathbf{x}$ and $\mathbf{y}$. For a given point $c \in (\mathbb{F}_q)^n$, we define:

$$\frac{\partial f}{\partial c} = \mathrm{D}f(c, \mathbf{x})$$

When f is quadratic, then $\mathrm{D}f$ is a symmetric bilinear function, and therefore $\frac{\partial f}{\partial c}$ is a linear mapping. By an abuse of notation, we also denote by $\frac{\partial f}{\partial c}$ the corresponding matrix.

A Structural Observation on IP Problems. Amongst the various parameters that influence the hardness of breaking instances of the IP problem, two play a special, correlated role: whether S and T are *linear* or *affine* transforms on the one hand, and whether the polynomials in $\mathbf{a}$ and $\mathbf{b}$ are *homogeneous* or not. Let us assume that $T(x) = T_c + T_\ell \cdot x$, and $S(x) = S_c + S_\ell \cdot x$, with $S_\ell, T_\ell \in \mathrm{GL}_n(\mathbb{F}_q)$. The following lemma is the fundamental result over which the Gröbner-based algorithm of [20] is built:

Lemma 1. *i) For all $k \geq 1$, we have:*

$$\mathbf{b}^{(k)} = T_\ell \circ \left(\mathbf{a} + \frac{\partial \mathbf{a}}{\partial S_c}\right)^{(k)} \circ S_\ell$$

ii) In particular, if S and T are linear, *then for all k we have:* $\mathbf{b}^{(k)} = T \circ \mathbf{a}^{(k)} \circ S$

Proof. See annex A.1.

If k denotes the total degree of $\mathbf{a}$, then $\left(\frac{\partial \mathbf{a}}{\partial x}\right)^{(k)} = 0$. The first point of lemma 1 then shows that the solution of an affine instance is also the solution of the linear homogeneous instance formed by taking only the homogeneous component of higher degree of $\mathbf{a}$ and $\mathbf{b}$.

Conversely, if the linear component of the solution of an affine instance is known, retrieving the affine part of the solution is often easy, and it can be accomplished using the technique shown in [27], which we recall. Lemma 1, point $i)$ shows that

$$T_\ell^{-1} \circ \mathbf{b}^{(1)} \circ S_\ell^{-1} - \mathbf{a}^{(1)} = \left(\frac{\partial \mathbf{a}}{\partial S_c}\right)^{(1)}. \tag{1}$$

Once S_ℓ and T_ℓ is known, then the left-hand side of (1) may be computed, and the right-hand side is a function of degree $d-1$ of S_c. In the particular case where $d = 2$, equation (1) in fact describes a system of linear equation that admits S_c as a solution.

A conclusion is that when the polynomials are quadratic, the linear homogeneous case is *complete*, and thus contains the hardest instances. This is in accordance with the experimental results of [20]: they found that their algorithm was polynomial on linear inhomogeneous instances, and exponential on linear homogeneous instances. This is also in accordance with the results presented in this paper: polynomial time algorithms are sometimes applicable to the inhomogeneous case (section 3.2 and 4), but often only exponential algorithms deal with the homogeneous case.

It should also be clear that S transforms the set of common zeroes of $\mathbf{a}$ into the set of common zeroes of $\mathbf{b}$. The consequences of this simple fact are discussed in sections 3.2 and 4.3.

A Meta-Algorithm For IP Problems. In this section, we present a meta-algorithm that can be applied to a wide variety of "IP-like" situations. To recover the isomorphism between two vectors of polynomials, we go through the following steps:

1. Find as many independent linear equations between the coefficients of S and those of T^{-1} as possible. Denote them by $\mathcal{S}$.
2. If $\mathcal{S}$ has rank $2n^2 - 1$, then both S and S^{-1} can be retrieved, and we are done.
3. Otherwise, (S, T^{-1}) lives in a vector space of dimension $k = \dim \ker \mathcal{S}$. Applying the algorithm of [20] yields a system $\mathcal{S}_{\text{quad}}$ of $un^2/2$ quadratic equations in k unknown.
4. Solve the system of quadratic equations by computing a Gröbner basis of $\mathcal{S}_{\text{quad}}$. This yields S.

This method yields all the possible solutions of a given instance. Moreover, it succeeds with probability one, and is deterministic. At the beginning of stage 3, S and T^{-1} can be expressed as a linear combinaison of k elements. It is expected that the complexity of the whole procedure is dominated by the last step, the complexity of which is difficult to predict in general. However, if all the equations of $\mathcal{S}_{\text{quad}}$ were linearly independent, and $k(k-1)/2 \leq u \cdot n^2/2$, then the equations of $\mathcal{S}_{\text{quad}}$ could be solved by linearization in time $\mathcal{O}\left(n^6\right)$ (and likely $\mathcal{O}\left(n^5\right)$ using more sophisticated sparse algebra subroutines). This is the first time in this paper that we point out that bigger values of u actually make the problem *easier*. In practice, if $\mathcal{S}_{\text{quad}}$ is that overdetermined, the computation of the Gröbner basis will also be polynomial (with roughly the same complexity).

The heart of the meta-algorithm therefore lies in collecting independent linear equations, and the main difficulty in analyzing its complexity is estimating the rank of $\mathcal{S}$. This general strategy will be instantiated twice in this paper: for the IP1S problem in section 3 and for a particular case of the full IP problem in section 4.2.

3 Isomorphism of Polynomials with One Secret

In this section, we investigate more particularly the IP1S problem; i.e. given two sets of polynomials $\mathbf{b}$ and $\mathbf{a}$ the task is to find S such that:

$$\mathbf{b} = \mathbf{a} \circ S. \tag{2}$$

A consequence of the structural observation of section 2 is that solving the linear homogeneous case is sufficient to solve all the other ones (this observation was also present in [42]). It was pointed out before that if there is only one quadratic polynomial, then the problem is easily solved in polynomial time (because quadratic forms admit a canonical representation, see [32]). We will therefore focus on the case of $u \geq 2$ quadratic polynomials, and on the case of one cubic polynomial. These problems are qualitatively pretty different, therefore we study them separately.

3.1 Quadratic IP1S

The quadratic IP1S problem is a good candidate for the application of the general strategy described in section 2. The important idea that we will use throughout this paper is that by *differentiating* equation (2), it will be possible to collect linear equations between the coefficients of S and those of S^{-1}. Indeed, for all vectors $\mathbf{x}, \mathbf{y} \in (\mathbb{F}_q)^n$, we have:

$$\forall \mathbf{x}, \mathbf{y} \in (\mathbb{F}_q)^n, \quad \mathrm{D}\mathbf{b}(\mathbf{x}, \mathbf{y}) = \mathrm{D}\mathbf{a}\left(S \cdot \mathbf{x}, S \cdot \mathbf{y}\right).$$

Using the change of variable $\mathbf{y}' = S \cdot \mathbf{y}$, this equation becomes:

$$\forall \mathbf{x}, \mathbf{y}' \in (\mathbb{F}_q)^n, \quad \mathrm{D}\mathbf{b}(\mathbf{x}, S^{-1} \cdot \mathbf{y}') = \mathrm{D}\mathbf{a}(S \cdot \mathbf{x}, \mathbf{y}'). \tag{3}$$

When $\mathbf{a}$ and $\mathbf{b}$ are of total degree 2, $\mathrm{D}\mathbf{a}$ and $\mathrm{D}\mathbf{b}$ are *bilinear* (symmetric) mappings. In this case, since equation (3) is valid for all $\mathbf{x}$ and $\mathbf{y}$, then in particular it is valid on a basis of $(\mathbb{F}_q)^n \times (\mathbb{F}_q)^n$, and substituting fixed basis vectors for $\mathbf{x}$ and $\mathbf{y}$ yields *linear equations* between the coefficients of S and those of S^{-1}. This way of presenting things generalizes an idea suggested in section 9 of [41], but tested only on toy examples that could be solved by hand, and not analyzed in a more general setting. We will show that the technique presented in [41] cannot by itself break reasonably-sized IP1S instances, but that it can be used successfully as the first step of our meta-algorithm. Obtaining linear equations can be described relatively simply using the usual theory of quadratic forms. This highlight the fact that the IP1S problem is a bit different in odd and even characteristic.

Quadratic IP1S in Odd Characteristic. If $\mathbb{F}_q$ is a field of odd characteristic, then the set of (homogeneous) quadratic polynomials is in one-to-one correspondance with the set of symmetric matrices. Let $\mathcal{P}(\mathbf{a}_k)$ (resp. $\mathcal{P}(\mathbf{b}_k)$) denote the matrix of the bilinear form associated with $\mathbf{a}_k$ (resp $\mathbf{b}_k$). This is sometimes called the polar form of $\mathbf{a}_k$. Recall that the coefficient of index (i,j) of $\mathcal{P}(\mathbf{a}_k)$ is $\mathrm{D}\mathbf{a}_k(e_i,e_j)/2$, where $(e_i)_{1\leq i\leq n}$ is a basis of $(\mathbb{F}_q)^n$. We then have:

$$S^{-1}\cdot\mathcal{P}(\mathbf{b}_k)=\mathcal{P}(\mathbf{a}_k)\cdot{}^tS \tag{4}$$

We always have at least two such equations. Assume that $\mathcal{P}(\mathbf{b}_1)$ and $\mathcal{P}(\mathbf{b}_2)$ were invertible. Then we obtain:

$$S^{-1}=\mathcal{P}(\mathbf{a}_1)\cdot{}^tS\cdot\mathcal{P}(\mathbf{b}_1)^{-1}$$
$$S^{-1}=\mathcal{P}(\mathbf{a}_2)\cdot{}^tS\cdot\mathcal{P}(\mathbf{b}_2)^{-1}$$

And therefore we obtain the equation given in [41]:

$$\mathcal{P}(\mathbf{a}_1)\cdot{}^tS\cdot\mathcal{P}(\mathbf{b}_1)^{-1}=\mathcal{P}(\mathbf{a}_2)\cdot{}^tS\cdot\mathcal{P}(\mathbf{b}_2)^{-1} \tag{5}$$

It is implicitly suggested in [41] that this is sufficient to recover S. We now move on the analysis of this technique, and more specifically we discuss the rank of the equations on S obtained this way. If $\mathcal{P}(\mathbf{b}_1)$ is invertible, then so is $\mathcal{P}(\mathbf{a}_1)$, and we get:

$${}^tS\cdot\mathcal{P}(\mathbf{b}_1)^{-1}\cdot\mathcal{P}(\mathbf{b}_2)=\mathcal{P}(\mathbf{a}_1)^{-1}\mathcal{P}(\mathbf{a}_2)\cdot{}^tS$$

Clearly, ${}^tS$ is the solution of the following matrix equation:

$$X\cdot\mathcal{P}(\mathbf{b}_1)^{-1}\cdot\mathcal{P}(\mathbf{b}_2)=\mathcal{P}(\mathbf{a}_1)^{-1}\cdot\mathcal{P}(\mathbf{a}_2)\cdot X$$

However, $\mathcal{P}(\mathbf{b}_1)$ and $\mathcal{P}(\mathbf{a}_1)$ are clearly related, which leads to:

$$\mathcal{P}(\mathbf{b}_1)^{-1}\cdot\mathcal{P}(\mathbf{b}_2)={}^tS^{-1}\cdot\mathcal{P}(\mathbf{a}_1)^{-1}\cdot\mathcal{P}(\mathbf{a}_2)\cdot{}^tS$$

And the previous matrix equation is equivalent to:

$$\left(X\cdot{}^tS^{-1}\right)\cdot\mathcal{P}(\mathbf{a}_1)^{-1}\cdot\mathcal{P}(\mathbf{a}_2)=\mathcal{P}(\mathbf{a}_1)^{-1}\mathcal{P}(\mathbf{a}_2)\cdot\left(X\cdot{}^tS^{-1}\right)$$

Therefore, there is a vector-space isomorphism between the set of solutions of (5) and the commutant of $\mathcal{P}(\mathbf{a}_1)^{-1}\cdot\mathcal{P}(\mathbf{a}_2)$, which we will denote by $\mathcal{C}$. Now, it is well-known that the dimension of $\mathcal{C}$ is lower-bounded by n (see [6, Fact 2.18.9], for instance). This brings us to a conclusion. By itself, the technique presented in [41] cannot solve the quadratic IP1S problem in odd characteristic, even when both quadratic forms are non-degenerate. The reason for this is that equation (5) never has less than q^n solutions.

However, the linear equation collected this way can be used in the context of our meta-algorithm described in section 2. For that, we would need to know an *upper-bound* on the dimension of $\mathcal{C}$. We address a slightly more general version of this problem below.

Quadratic IP1S in Characteristic Two. The case where $\mathbb{F}_q$ is a field of characteristic two is more interesting from the cryptographic point of view, and more challenging. The theory of quadratic forms is completely different in characteristic two, and in particular there is no longer a canonical representation of quadratic forms by symmetric matrices that would yield convenient linear relations. This is fact not a problem, as we may still define $\mathcal{P}(\mathbf{a}_k)_{i,j}=\mathrm{D}\mathbf{a}_k(e_i,e_j)$, and equation (4) still holds.

Towards a Complexity Analysis. In both case, we obtain u matrix equations following the pattern of equation (4). The rank of the equations between the individual coefficients of S and S^{-1} is therefore clearly an increasing function of u. For this reason, and because of the previous observation that big values of u make the last step of the attack easier, we will focus on the hardest case where $u=2$. The problem we are facing now is to evaluate the rank of the following system of homogeneous linear equations:

$$\mathcal{S}:\quad\begin{cases}S^{-1}\cdot\mathcal{P}(\mathbf{b}_1)=\mathcal{P}(\mathbf{a}_1)\cdot{}^tS\\ S^{-1}\cdot\mathcal{P}(\mathbf{b}_2)=\mathcal{P}(\mathbf{a}_2)\cdot{}^tS\end{cases}$$

This system is made of $2n^2$ linear equations in $2n^2$ unknowns (the coefficients of S and S^{-1}). What makes its study a bit delicate is that $\mathcal{P}(\mathbf{b}_k)$ and $\mathcal{P}(\mathbf{b}_k)$ are not at all statistically independent. For instance, we know that $\mathcal{S}$ admits a non-trivial solution, which is precisely (S,S^{-1}). Therefore all the equations *cannot* be linearly independent, and the rank of $\mathcal{S}$ has to be studied carefully.

Practical Behavior of the Algorithm. As a foreword, we would like to point out that the rank of $\mathcal{S}$ is usually higher in practice than what the analysis suggests. Therefore, the algorithm empirically works better than what the analysis suggests. For the sake of simplicity, the analysis presented below only works under some hypotheses. For instance, we require one of the two quadratic forms to be non-degenerate, and this happens with a non-negligible probability in the random case. However, if both quadratic forms are of rank, say, $n-1$, then the algorithm still works pretty well, even though our analysis says nothing about it.

For this reason, we would like to expose the practical behavior of the algorithm before going into the detail of its complexity analysis. We measured the actual rank of $\mathcal{S}$ with $u = 2$ for various values of q and n, and we observed that for bigger fields (say $\mathbb{F}_{16}$), the dimension of the kernel of $\mathcal{S}$, denoted by k, is $2n$ all the time. Over $\mathbb{F}_2$, accidental cancellation of coefficients is more likely, and we have measured:

n	$\mathbb{P}[k = 2n]$	$\mathbb{P}[k = 2n+4]$	$\mathbb{P}[k = 2n+8]$	$\mathbb{P}[k = 2n+12]$
8	0.88	0.05	0.05	0.02
16	0.96		0.04	
32	0.94		0.06	
64	0.96		0.04	

The algorithm has been implemented using the computer algebra system MAGMA [8]. Solving the equations of $\mathcal{S}_{\text{quad}}$ is achieved by first computing a Gröbner basis of these equations for the Graded-Reverse Lexicographic order using the F4 algorithm [16], and then converting it to the Lexicographic order using the FGLM algorithm [15]. The implementation breaks all the proposed quadratic IP1S challenges in negligible time, less than 30 seconds for the biggest one. To illustrate its effectiveness, we built and broke a random IP1S instance with $q = 2$, $u = 2$ and $n = 32$ in a matter of seconds (recall that the biggest proposed quadratic IP1S challenge, and never broken before was $q = u = 2$ and $n = 16$). The dominating part in the execution of the algorithm is in fact the symbolic manipulation of polynomials required to write down the equations of $\mathcal{S}_{\text{quad}}$. Actually solving the resulting quadratic equations turns out to be easier than generating them.

Analysis of the Rank of $\mathcal{S}$. We now undertake the task of theoretically justifying the empirical success of the algorithm. The main parameter influencing its complexity is the number of independent linear equations in $\mathcal{S}$.

To make the analysis simpler, we will assume that $\mathbf{a}_1$ (and therefore $\mathbf{b}_1$) is non-degenerate. This is equivalent to saying that $\mathcal{P}(\mathbf{a}_1)$ is invertible. We discuss below the probability that this event is realized.

We are now ready to state the technical result that underlies our analysis. This theorem will use well-known result about the Smith Normal Form of a characteristic matrix and its invariants. Given a square matrix M, the non-unit polynomials $p_1, \ldots p_s$ on the diagonal of the Smith Normal Form of the characteristic $M - x1_n$ are the *invariant factors* of M. Their product is the characteristic polynomial of M, p_s is the minimal polynomial of M, and p_i divides p_{i+1}.

Theorem 1. *Let A_1, A_2, B_1, B_2 be four given matrices of size $n \times n$ with coefficients in $\mathbb{F}_q$. Let us consider the set of all pairs (X, Y) of $n \times n$ matrices satisfying the following linear equations:*

$$\mathcal{S} : \quad \begin{cases} X \cdot B_1 - A_1 \cdot Y = 0 \\ X \cdot B_2 - A_2 \cdot Y = 0 \end{cases}$$

If A_1 and B_1 are invertible, we define:

$$\begin{aligned} M_A &= \quad A_1^{-1} \cdot A_2 \\ M_B &= -\, {}^t\left(B_1^{-1} \cdot B_2\right) \end{aligned}$$

Let $p_1, \ldots, p_s$ denote the invariant factors of M_B. We then have:

$$\operatorname{rank} \mathcal{S} = 2n(n-s) + \sum_{i=1}^{s} \operatorname{rank}\, p_i\left(M_A\right)$$

Proof. See annex A.2.

The analysis below estimates the probability of success of the attack by computing the probability that A_1 is invertible (that depends on the field characteristic) and we estimate the probability that the rank of $\mathcal{S}$ is $2n^2 - 2n$ by estimating the probability that $s = 1$ or 2. This is achieved by using a result on the probability that the characteristic polynomial of a random matrix be its minimal polynomial.

Application of theorem 1 to the Random Case. Theorem 1 holds only if $\mathcal{P}\left(\mathbf{b}_1\right)$ or $\mathcal{P}\left(\mathbf{b}_2\right)$ is invertible (we may swap them if we wish). If q is odd, then $\mathcal{P}\left(\mathbf{b}_1\right)$ is a random symmetric matrix, and the probability that it has a given rank is given by lemma 7 (found in annex B). The probability that $\mathcal{P}\left(\mathbf{b}_1\right)$ is invertible is about 0.639 for $q = 3$, and is a (rapidly) increasing function of q. The probability that either $\mathcal{P}\left(\mathbf{b}_1\right)$ or $\mathcal{P}\left(\mathbf{b}_2\right)$ is invertible is about 0.870 for $q = 3$. If q is even, then $\mathcal{P}\left(\mathbf{b}_1\right)$ is a random symmetric matrix with zeros on the diagonal, and the probability that it has a given rank is given by lemma 8 (also found in annex B). The probability that $\mathcal{P}\left(\mathbf{b}_1\right)$ is invertible if $q = 2$ is about 0.419 (again, this probability increase exponentially with q). The probability that either $\mathcal{P}\left(\mathbf{b}_1\right)$ or $\mathcal{P}\left(\mathbf{b}_2\right)$ is invertible is about 0.662 for $q = 2$.

Theorem 1 is then applicable in more than half of the cases. When it is applicable, what guarantee does it exactly offer? We would need to know something about the invariant factors of M_B. It is shown in [43, proposition 12] that the expected number of invariant factors (counting multiplicities) in a random invertible matrix over $\mathbb{F}_q$ is $\log n + \mathcal{O}(1)$. An easy case would be when the minimal and characteristic polynomials are the same (then there is only one invariant factor, and it is χ_{M_B}). The probability of this event is given by lemma 9 (found in annex B): for random matrices over $\mathbb{F}_2$, and for n big enough, the proportion of cyclic matrices approaches 0.746. Unfortunately, in even characteristic, M_B is *never* cyclic: we observed (but did not succeed to prove) that if U and V are zero-diagonal symmetric matrices over $\mathbb{F}_2$, χ_{UV} is a square, and that $\sqrt{\chi_{UV}}$ is an annihilator of UV. The minimal polynomial of UV is therefore of degree at most $n/2$: the product of two zero-diagonal symmetric matrices over $\mathbb{F}_2$ is thus far from behaving randomly.

As mentioned above, UV cannot have only one invariant factor (because this would mean that the minimal and characteristic polynomials are the same, which is not the case). We measured that UV has exactly two invariant factors (both being $\sqrt{\chi_{UV}}$) with probability about 0.74 for $q = 2$ (this probability seems to increase with q and n, and is strikingly close to the proportion of cyclic random matrices...). In this setting, theorem 1 guarantees that the dimension of $\ker \mathcal{S}$ is exactly $2n$ (because M_A and M_B are similar, therefore the minimal polynomial of M_B vanishes on M_A). This is not completely sufficient to guarantee that we will be able to solve the equations of $\mathcal{S}_{\text{quad}}$ by linearization (because we would have $2n^2$ equations in $4n^2$ variables). However, it creates a system of quadratic equations so overdefined that the maximal degree of polynomials reached when computing a Gröbner basis appears to be always upper-bounded by 2 in practice. The computation of the Gröbner basis therefore terminates in time $\mathcal{O}\left(n^6\right)$ in most cases.

3.2 Cubic IP1S

In this section, we focus on the case where $\mathbf{a}$ and $\mathbf{b}$ are composed of a single cubic polynomial. We assume that $\mathbf{a}$ and $\mathbf{b}$ are given explicitly, i.e.:

$$\mathbf{a} = \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} A_{i,j,k} \cdot x_i x_j x_k, \qquad \mathbf{b} = \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} B_{i,j,k} \cdot x_i x_j x_k.$$

As already explained, we can restrict our attention to the homogenous case. The techniques developed previously for the quadratic case cannot directly applied in this setting. Indeed, the differential is no longer a bilinear mapping, and then there is no obvious linear equations between the coefficients of a solution and those of its inverse. However, we can combine the use of the differential together with the Gröbner basis approach proposed in [20]. We will denote by $S_0 = \{s_{i,j}^0\}_{1 \leq i,j \leq n}$ a particular solution of IP1S between $\mathbf{a}$ and $\mathbf{b}$, i.e. it holds that $\mathbf{b} = \mathbf{a} \circ S_0$. For all vectors $\mathbf{x}, \mathbf{y} \in \left(\mathbb{F}_q\right)^n$, we have:

$$\mathrm{D}\mathbf{a}(S_0 \cdot \mathbf{x}, \mathbf{y}) = \mathrm{D}\mathbf{b}(\mathbf{x}, S_0^{-1} \cdot \mathbf{y}).$$

$\mathbf{a}$ and $\mathbf{b}$ being of total degree 3, the coefficients of S_0 and S_0^{-1} appear with degree two in the expression of $\mathrm{D}\mathbf{a}$ and $\mathrm{D}\mathbf{b}$ above. Let R be the ring $\mathbb{K}[s_{1,1}, \ldots, s_{n,n}, u_{1,1}, \ldots, u_{n,n}]$. We consider the algebra $\mathcal{A}^s$ of all $n \times n$ matrices over the ring R. Let $S = \{s_{i,j}\}$ and $U = \{u_{i,j}\}$ in $\mathcal{A}^s$ be symbolic matrices. We denote by $\mathcal{I}_{\mathbf{a},\mathbf{b}}$ the ideal generated by all the coefficients in R of the equations:

$$\mathrm{D}\mathbf{a}(S \cdot \mathbf{x}, \mathbf{y}) - \mathrm{D}\mathbf{b}(\mathbf{x}, U \cdot \mathbf{y}) = 0, \qquad U \cdot S - 1_n = 0_n, \qquad S \cdot U - 1_n = 0_n.$$

It is easy to see that $U = S_0^{-1}$ and $S = S_0$ is particular solution of this system, and also a solution of IP1S between $\mathbf{b}$ and $\mathbf{a}$.

Our goal is to provide an upper bound on the maximum degree reached during a Gröbner basis computation of $\mathcal{I}_{\mathbf{a},\mathbf{b}}$. This degree, called *degree of regularity*, is the key parameter in establishing the complexity of the Gröbner basis computation. Indeed, the cost of computing a Gröbner basis is polynomial in the degree of regularity D_{reg} of the system[4] considered: $\mathcal{O}(N^{\omega \cdot D_{\mathrm{reg}}})$, with $2 < \omega \leq 3$ the linear algebra constant, and N the number of variables of system considered. In our case, $N = n^2$.

The behavior of the degree of regularity is well understood for "random" systems of equations [3,4,5] (*i.e.*, regular or semi-regular systems). On the other hand, as soon as the system has some kind of structure, which is always the case in cryptography, this degree is much more difficult to predict. In some particular cases, it is however possible to bound the degree of regularity (see for HFE [19,29]). We prove here that $D_{\mathrm{reg}} = 2$ for $\mathcal{I}_{\mathbf{a},\mathbf{b}}$ under the hypothesis that we know one row of a particular solution S_0. We assume then that we know the following ideal:

$$\mathcal{J} = \left\langle s_{1,j} - s_{1,j}^{(0)} \mid j = 1, \ldots, n \right\rangle .$$

Theorem 2. *The degree of regularity of the ideal $\mathcal{I}_{\mathbf{a},\mathbf{b}} + \mathcal{J}$ is 2. Therefore, computing a Gröbner basis of this ideal takes time $\mathcal{O}\left(n^6\right)$.*

Proof of theorem 2. We use the fact that the degree of regularity of an ideal is generically left invariant by any linear change of the variables or generators [31]. In particular, we consider the ideal $\mathcal{I}'_{\mathbf{a},\mathbf{b}}$ generated by all the coefficients in $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_n]$ of the equations:

$$\mathrm{D}\mathbf{a}(S_0(S + I_n)\mathbf{x}, \mathbf{y}) - \mathrm{D}\mathbf{b}(\mathbf{x}, (U + I_n)S_0^{-1}\mathbf{y}) = 0, \qquad U \cdot S = 0_n, \qquad S \cdot U = 0_n.$$

It is clear that $\mathcal{I}'_{\mathbf{a},\mathbf{b}}$ is obtained from $\mathcal{I}_{\mathbf{a},\mathbf{b}}$ by replacing S (resp. U) by $(I_n + S_0)S$ (resp. $(U + I_n)S_0^{-1}$). Thus, the degree of regularity of $\mathcal{I}'_{\mathbf{a},\mathbf{b}}$ and $\mathcal{I}_{\mathbf{a},\mathbf{b}}$ are equal. Using the same transformation, the ideal $\mathcal{J}$ becomes

$$\mathcal{J}' = \langle s_{1,j} \mid j = 1, \ldots, n \rangle .$$

We now estimate the degree of regularity of the ideal $\mathcal{I}'_{\mathbf{a},\mathbf{b}} + \mathcal{J}'$. For a reason which will become clear in the sequel, it is more convenient to work with $\mathcal{I}'_{\mathbf{a},\mathbf{b}} + \mathcal{J}'$. In what follows, F will denote the generators of $\mathcal{I}'_{\mathbf{a},\mathbf{b}} + \mathcal{J}'$. We will show that many new linear equations appear when considering equations of degree 2. To formalize this, we introduce some definitions related to the F_4 algorithm to compute Gröbner bases [17]. In particular, we will denote by $I_{d,k}$ the linear space generated during the k-th step of F_4 when considering polynomials of degree d.

Definition 1. *We have the following recursive definition of $I_{d,k}$:*

$$\begin{aligned}
I_{d,0}(F) &= \mathit{Vect}_{\mathbb{K}}\left(F\right) \\
I_{d,1}(F) &= \mathit{Vect}_{\mathbb{K}}\left(s_{i,j} f \ \mid \ 1 \leqslant i,j \leqslant n \ \mathit{and} \ f \in I_{d,0}(F)\right) \\
&\quad + \mathit{Vect}_{\mathbb{K}}\left(u_{i,j} f \ \mid \ 1 \leqslant i,j \leqslant n \ \mathit{and} \ f \in I_{d,0}(F)\right) \\
I_{d,k}(F) &= \mathit{Vect}_{\mathbb{K}}\left(s_{i,j} f \ \mid \ 1 \leqslant i,j \leqslant n \ \mathit{and} \ f \in I_{d,k-1}(F) \ \mathit{and} \ \deg(f) \leq d-1\right) \\
&\quad + \mathit{Vect}_{\mathbb{K}}\left(u_{i,j} f \ \mid \ 1 \leqslant i,j \leqslant n \ \mathit{and} \ f \in I_{d,k-1}(F) \ \mathit{and} \ \deg(f) \leq d-1\right).
\end{aligned}$$

Roughly speaking, the index k is the number of steps in the F_4/F_5 [18] algorithm to compute an element $f \in I_{d,k}(F)$. We will show that $I_{2,1}(F)$ contains exactly $n^2 + 2n$ linear equations. This means that we have already many linear equations generated during the first step of the computation of a Gröbner basis of F.

Lemma 2. *$I_{2,1}(F)$ contains the following linear equations:*

$$\{u_{1,j} \mid j = 1, \ldots, n\}. \tag{6}$$

Proof. From the first row of the following zero matrix $S \cdot U$ we obtain the following equations:

$$\begin{cases}
s_{1,1}\, u_{1,1} + s_{1,2}\, u_{2,1} + s_{1,3}\, u_{3,1} + \cdots + s_{1,n}\, u_{n,1} = 0, \\
s_{1,1}\, u_{1,2} + s_{1,2}\, u_{2,2} + s_{1,3}\, u_{3,2} + \cdots + s_{1,n}\, u_{n,2} = 0, \\
s_{1,1}\, u_{1,3} + s_{1,2}\, u_{2,3} + s_{1,3}\, u_{3,3} + \cdots + s_{1,n}\, u_{n,3} = 0, \\
\quad \cdots \\
s_{1,1}\, u_{1,n} + s_{1,2}\, u_{2,n} + s_{1,3}\, u_{3,n} + \cdots + s_{1,n}\, u_{n,n} = 0
\end{cases}$$

[4] This is true in the zero-dimensionnal case, *i.e.*, when the number of solutions is finite.

Using the equations $s_{1,j} = 0$ from the ideal $\mathcal{J}'$, we obtain then $u_{1,1} = 0, u_{1,2} = 0, \ldots, u_{1,n} = 0$. □

We can also predict the existence of other linear equations in $I_{2,1}(F)$.

Lemma 3. *For all $(i, j) \in \{1, \ldots, n\}^2$ the coefficient of $y_1 y_i x_j$ in $\mathrm{D}\mathbf{a}(S_0(S + I_n)\mathbf{x}, \mathbf{y}) - \mathrm{D}\mathbf{b}(\mathbf{x}, (U + I_n)S_0^{-1}\mathbf{y})$ is a non zero*[5] *linear equation modulo the equations of the ideal $\mathcal{J}'$ and* (6)*. Among these equations, there are n which depend only of the variables $\{s_{k,\ell} \mid 1 \leq k, \ell \leq n\}$.*

Proof. See Annex A.3.

To summarize:

Lemma 4. *$I_{2,1}(F)$ contains exactly $n^2 + 2n$ linear equations.*

Proof. In $I_{2,1}(F)$, we have: n linear equations from Lemma 3, n linear equations from the very definition of $\mathcal{J}'$, and n^2 linear equations from lemma 3 □

As explained before, we obtain $n^2 + 2n$ linear equations for $I_{2,1}(F)$. However, we have $2n^2$ variables. So, we have to consider $I_{2,2}(F)$, *i.e.*, the equations generated at degree 2 during the second step. Thanks to lemma 4, we can reduce the original system to a quadratic system in $2n^2 - (2n + n^2) = (n - 1)^2$ variables. W.l.o.g we can assume that we keep only the variable $u_{i,j}$ where $2 \leq i, j \leq n$. Let F' be the system obtained from F after substituting the $2n + n^2$ linear equations of lemma 4. All the monomials in $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_n]$ of $\mathrm{D}\mathbf{a}(S_0(S + I_n)\mathbf{x}, \mathbf{y}) - \mathrm{D}\mathbf{b}(\mathbf{x}, (U + I_n)S_0^{-1}\mathbf{y})$ have the following shape:

$$x_i y_j y_k \text{ or } y_i x_j x_k \text{ with } 1 \leq i, j, k \leq n.$$

Hence the number of such monomials is $2n\frac{n(n+1)}{2} = n^2(n + 1) \approx n^3$, which implies that the number of equations in F' is also n^3. Thanks to this remark, we will now prove that we can linearize F'. Let $T(F')$ be the set of all monomials occurring in F'. We can assume that $T(G') = [t_1 < t_2 < \cdots < t_N]$. It is important to remark that $t_1 = u_{2,2}$ up to $t_{(n-1)^2} = u_{n,n}$ are in fact variables. Now, let M be the matrix representation of G' w.r.t. $T(G')$. Since we know precisely the shape of the equations from the proof of lemma 3, it is possible to establish that:

1. most of the equations are very sparse, namely each equation contains about n^2 non-zero terms.
2. all the variables $t_1, \ldots, t_{(n-1)^2}$ occur in *all* the equations

After a Gaussian elimination of the matrix M, we obtain the following shape:

$$\widetilde{M} = \begin{bmatrix} 1_{(n-1)^2} & 0 & 0 & 0 \\ 0 & \times & \cdots & \cdots \\ 0 & \times & \ddots & \vdots \\ 0 & \times & \cdots & \ddots \end{bmatrix}$$

Hence, we obtain after a second step of computation in degree 2 the equations $u_{2,2} = \cdots = u_{n,n} = 0$. This means that after 2 steps of computation at degree 2, we obtain $(n-1)^2 + 2n + n^2 = 2n^2$ linear equations in $2n^2$. This explains why the maximum degree reached during the Gröbner basis computation of $\mathcal{I}'_{\mathbf{a},\mathbf{b}} + \mathcal{J}'$ is bounded by 2, and concludes the proof of theorem 2.

Application to the Linear Inhomogeneous Case. If S can be assumed to be a linear bijective mapping, and if $\mathbf{a}$ has a non-trivial homogeneous component of degree 1, then we are in a situation where theorem 2 is applicable, and S can be determined though a Gröbner basis computation which terminates in time $\mathcal{O}\left(n^6\right)$.

[5] more precisely, generically non zero.

Application to the Other Cases. All the other cases reduce to the linear homogeneous case, as mentionned in section 2. In this setting, the problem is that we do not have enough knowledge on S to make the Gröbner basis computation efficient. A simple idea would be to guess a colum of S then compute the Gröbner basis. This approach has complexity $\mathcal{O}\left(n^6 q^n\right)$.

We discuss a possible idea to improve this complexity. We already mentionned that S sends the zeroes of $\mathbf{a}$ onto those of $\mathbf{b}$. Let us denote by $Z_{\mathbf{a}}$ (resp. $Z_{\mathbf{b}}$) the set of zeroes of $\mathbf{a}$ (resp. $\mathbf{b}$). We have $S(Z_{\mathbf{a}}) = Z_{\mathbf{b}}$. In particular, by the linearity of S we have:

$$S\left(\sum_{\mathbf{x}\in Z_{\mathbf{a}}} \mathbf{x}\right) = \sum_{\mathbf{y}\in Z_{\mathbf{b}}} \mathbf{y}$$

This yields a relation on S, which is enough to use theorem 2. $\mathbf{a}$ and $\mathbf{b}$ may be assumed to have about q^{n-1} zeroes. Finding them requires time $\mathcal{O}\left(q^n\right)$. The complexity of the attack could thus be improved to $\mathcal{O}\left(n^6 + q^n\right)$. Unfortunately, this trick fails systematically, as a consequence of the Chevalley-Warning theorem.

Lemma 5. *The sum of the zeroes of a cubic form on 5 variables or more over $\mathbb{F}_q$ is always zero.*

Proof. Let us consider the elements of $Z_{\mathbf{a}}$ having α as their first coordinate, and let us denote by n_α their number. These are in fact the common zeroes of $(\mathbf{a}, x_1 - \alpha)$. By the Chevalley-Warning theorem [10,44], if $\mathbf{a}$ has at least 5 variables, then the characteristic of the field divides n_α. Therefore, their sum has zero on the first coordinate. Applying this result for all values of α shows that the sum of zeroes of $\mathbf{a}$ has a null first coordinate. We then just consider all coordinates successively. □

4 Isomorphisms of Polynomials with Two Secrets

In this section we focus on the full IP problem (with two secret affine or linear mappings S and T). Recall that finding a polynomial isomorphism between two vectors of multivariate polynomials $\mathbf{a}$ and $\mathbf{b}$ means finding two matrices S_ℓ and T_ℓ, as well as two vectors S_c and T_c such that:

$$\mathbf{b}\left(\mathbf{x}\right) = T_c + T_\ell\left(\mathbf{a}\left(S_\ell \cdot x + S_c\right)\right) \tag{7}$$

Similarly to the IP1S problem, the hardness of the unrestricted IP problem depends on the parity of the characteristic of the field. As opposed to the IP1S problem though, it also depends pretty much on whether $u = n$, whether the polynomials are homogeneous and whether S and T are linear or affine. We first present two polynomial algorithms that solve the easiest case, namely the linear homogeneous IP problem with as many polynomial as variables, for which the Gröbner-based algorithm of [20] is also polynomial. While the latter runs in time $\mathcal{O}\left(n^9\right)$, a new heuristic algorithm runs in time $\mathcal{O}\left(n^3\right)$, and a more rigorous variation runs in time between $\mathcal{O}\left(n^4\right)$ and $\mathcal{O}\left(n^6\right)$.

4.1 A Fast Heuristic for the Linear Inhomogeneous Case

We first present a simple heuristic which combine the structural observation of section 2 and the to-and-fro approach [41]. As before, we will denote by $\mathbf{a}^{(1)}$ the homogeneous component of degree one of $\mathbf{a}$ (*i.e.*, the linear terms of $\mathbf{a}$). The application of lemma 1 immediately yields:

$$\mathbf{b}^{(1)} = T_\ell \cdot \mathbf{a}^{(1)} \cdot S_\ell \tag{8}$$

If $\mathbf{a}^{(1)}$ and $\mathbf{b}^{(1)}$ are invertible and if $\mathbf{a}(0) \neq 0$, then equation (8) allows us to use a very efficient variation of the to-and-fro method, which is shown in figure 1. The complexity of this algorithm is $\mathcal{O}\left(n^3\right)$. Inverting both $\mathbf{a}^{(1)}$ and $\mathbf{b}^{(1)}$ can be done once for all. The matrix-vector products take $\mathcal{O}\left(n^2\right)$ and there are n of them. Lastly, reconstructing S_ℓ and T_ℓ takes only $\mathcal{O}\left(n^3\right)$, because in the basis $\left(x_i\right)_{i\leq n}$, S is made of the y_i's. Changing the basis amounts to performing one matrix inversion and two matrix-matrix products.

This heuristic works well for random inhomogeneous instances (*i.e.*, instances where all the coefficients of all degrees are randomly chosen). It is straightforward that the number of $n \times n$ invertible matrices over $\mathbb{F}_q$ is $\prod_{i=0}^{n-1}\left(q^n - q^i\right)$. This tells us that the probability that $\mathbf{a}^{(1)}$ is invertible is about 0.288 (for $q = 2$, higher for bigger q) and the probability that $\mathbf{a}(0) \neq 0$ is $1 - 1/q^n$. Again, $q = 2$ looks like a worst case.

Fig. 1 A variant of the “To-and-Fro” IP algorithm using linear terms to go back

```
$x_1 \leftarrow \mathbf{a}(0)$
$y_1 \leftarrow \mathbf{b}(0)$
for $i = 1$ to $n$ do // At this point one has $y_i = T_\ell \cdot x_i$
    $y'_i \leftarrow \left(\mathbf{a}^{(1)}\right)^{-1} \cdot x_i$
    $x'_i \leftarrow \left(\mathbf{b}^{(1)}\right)^{-1} \cdot y_i$
    // And we obtain $y'_i = S_\ell \cdot x'_i$
    $y_{i+1} \leftarrow \mathbf{b}\left(x'_i\right)$
    $x_{i+1} \leftarrow \mathbf{a}\left(y'_i\right)$
end for
Reconstruct $S_\ell$ from the pairs $(x'_i, y'_i)$ and $T_\ell$ from the pairs $(x_i, y_i)$.
```

For all realistic sets of parameters, the heuristic either fails or terminates in less than a second, even for parameters that were taking several minutes to the Gröbner-based algorithm of [20].

While this technique works well on random instances, it is not very difficult to cook an instance on which it fails completely. The heuristic is not very “robust”, in the sense that not only it requires $\mathbf{a}^{(1)}$ and $\mathbf{b}^{(1)}$ to be invertible and $\mathbf{a}(0) \neq 0$, but it may also fail unexpectedly if $\mathbf{a}$ and $\mathbf{b}$ are not “random” enough. Indeed, the non-linearity of $\mathbf{a}$ and $\mathbf{b}$ plays a crucial role in making each new relation on S_ℓ and T_ℓ linearly independent from the previous ones, which is required for the algorithm to work.

Here is an example over which the heuristic fails. Suppose that $\mathbf{a} = T_\ell \circ \mathbf{a} \circ S_\ell$. Such an example in fact came up naturally when studying a variation of HFE (paper to appear). Our initial “bootstrapping” relation $T_\ell \cdot \mathbf{a}(0) = \mathbf{a}(0)$ in fact describes an eigenvector $\mathbf{x}$ satisfying the equation $T_\ell \cdot \mathbf{x} = \mathbf{x}$. Thanks to equation (8), this relation is transferred to an eigenvector of S_ℓ satisfying the equation $S_\ell \cdot \mathbf{y} = \mathbf{y}$ (with $\mathbf{y} = {\mathbf{a}^{(1)}}^{-1} \cdot \mathbf{x}$). Now, using the easy and natural way to produce new relations on T_ℓ, we obtain other eigenvectors of T_ℓ satisfying the same eigenvalue equation: $T_\ell \cdot \mathbf{a}(\mathbf{y}) = \mathbf{a}(\mathbf{y})$.

The number of independent linear relations that we may accumulate this way is upper-bounded by the dimension of the eigenspace of T_ℓ for the eigenvalue 1. This eigenspace cannot span the whole $(\mathbb{F}_q)^n$ (otherwise T_ℓ would be the identity matrix...). The heuristic thus cannot fully determine T_ℓ, as after a given point, all the new linear relations found at each iteration will be linearly dependent from the previous ones.

4.2 A More Robust Algorithm

To overcome this situation, we propose a new algorithm based on the meta-strategy that was successful with IP1S, namely obtaining linear equations by differentiating equation (7), and then solving the system of quadratic equations resulting obtained through the use of the Gröbner-based algorithm of [20] via linearization. This yields a more “robust”, less heuristic, slightly less efficient algorithm. The starting point is the fact that equation (8) provides n^2 linear relations between the coefficients of S_ℓ and those of T_ℓ^{-1}. This is not enough to recover the two matrices by linear algebra, since there are $2n^2$ unknowns. However, we will now show that we can obtain an additional matrix equality thanks to the differential, yielding enough linear equations. Differentiating equation (7) yields:

$$\mathrm{D}\mathbf{b}(\mathbf{x}, \mathbf{y}) = T_\ell \cdot \mathrm{D}\mathbf{a}(S_\ell \cdot \mathbf{x}, S_\ell \cdot \mathbf{y}) \tag{9}$$

Reusing the notations of the algorithm of figure 1, we have: $x'_1 = \left(\mathbf{b}^{(1)}\right)^{-1} \cdot \mathbf{b}(0)$ and $y'_1 = \left(\mathbf{a}^{(1)}\right)^{-1} \cdot \mathbf{a}(0)$, so that $S \cdot x'_1 = y'_1$. Fixing $x = x'_1$ in equation (9) gives:

$$\frac{\partial \mathbf{b}}{\partial x'_1} = T_\ell \cdot \frac{\partial \mathbf{a}}{\partial y'_1} \cdot S_\ell \tag{10}$$

which provides the n^2 new linear equations between the coefficients of S_ℓ and T_ℓ^{-1} that were required. Note that this technique requires a known relation on S_ℓ, which in turn requires that $\mathbf{a}(0) \neq 0$ *and* that $\mathbf{a}(0)$ belongs to the range of $\mathbf{a}^{(1)}$.

Analysis. Again, analyzing this algorithm boils down to estimate the rank of the following system of linear equations:

$$\mathcal{S}: \quad \begin{cases} T_\ell^{-1} \cdot \mathbf{b}^{(1)} = \mathbf{a}^{(1)} \cdot S \\ T_\ell^{-1} \cdot \frac{\partial \mathbf{b}}{\partial x_1'} = \frac{\partial \mathbf{a}}{\partial y_1'} \cdot S \end{cases} \tag{11}$$

This task is again non-trivial, and we will establish that the rank of $\mathcal{S}$ is *at most* $2n^2 - n$, even if all the matrices above are invertible. The matrix $\mathbf{b}^{(1)}$ is completely random, thus it is invertible with probability $\prod_{i=0}^{n-1} \left(1 - q^{i-n}\right)$ (greater than 0.288 for $q = 2$). Because of lemma 1, we know that $\mathbf{b}^{(1)}$ is invertible if and only if $\mathbf{a}^{(1)}$ is. The case of $\frac{\partial \mathbf{b}}{\partial x_1'}$ is a bit more subtle. In odd characteristic, it can be shown that is a random matrix, and thus the probability that it is invertible is also known. In characteristic two, we have $\frac{\partial \mathbf{b}}{\partial x_1'} \cdot x_1' = 0$. Thus $\frac{\partial \mathbf{b}}{\partial x_1'}$ is not invertible, and is not *a priori* random. The application of theorem 1 yields that the rank of $\mathcal{S}$ is $2n(n-s) + \sum_{i=1}^{s} \operatorname{rank} \, p_i(M_A)$, where:

$$M_A = \left(\mathbf{a}^{(1)}\right)^{-1} \cdot \frac{\partial \mathbf{a}}{\partial y_1'} \qquad\qquad M_B = -^t\left(\left(\mathbf{b}^{(1)}\right)^{-1} \cdot \frac{\partial \mathbf{b}}{\partial x_1'}\right)$$

and the p_i are the invariant factors of M_B. It must be noted that once more M_A and M_B are similar, which shows that the dimension of $\ker \mathcal{S}$ is lower-bounded by n (because p_s is the minimal polynomial of M_B, which also vanish on M_A). Now, the rank of $\mathcal{S}$ will be exactly $2n^2 - n$ if and only if the minimal and characteristic polynomial of M_B are the same. We already mentioned that this happens with a noticeable probability for random matrices, but unfortunately M_B is not random, as it always has a non-trivial kernel. However, the following lemma tells us that M_B is in fact random amongst the set of linear mappings that vanish at x_1'.

Lemma 6 ([14], theorem 2). *Let $\mathbb{F}_q$ be a field of characteristic two. Given a random quadratic map $f : (\mathbb{F}_q)^n \rightarrow (\mathbb{F}_q)^n$, and a non-zero element $\mathbf{x}$ of $(\mathbb{F}_q)^n$, the rank of $\frac{\partial f}{\partial \mathbf{x}}$ follows the distribution of ranks of endomorphisms of $(\mathbb{F}_q)^n$ vanishing at $\mathbf{x}$. Therefore, for any t in $[1, n]$ the probability that $\frac{\partial f}{\partial \mathbf{x}}$ has rank $n - t$ is:*

$$\frac{\lambda(n-1)\lambda(n)}{\lambda(n-t)\lambda(t-1)\lambda(t)} \cdot q^{(-t)(t-1)} \qquad \textit{with } \lambda(k) = \prod_{i=1}^{k} \left(1 - \frac{1}{q^i}\right)$$

Let $e_n = x_1'$, and let us extend it to a basis $e_1, \ldots, e_n$ of $(\mathbb{F}_q)^n$. In this new basis, the last column of M_B is zero. Expanding the determinant of $M_B - x \cdot 1_n$ along the last column yields $\chi_{M_B} = x \cdot \chi_{M_B'}$, where M_B' is the matrix obtained by removing the last row and the last column of the matrix representing the same endomorphism as M_B in the new basis. This shows that the characteristic polynomial of M_B is the product of x and of the characteristic polynomial of a random matrix of dimension $(n-1) \times (n-1)$. This time, since M_B' is random, lemma 9 is applicable and guarantees that the minimal polynomial of M_B' is of degree $n - 1$ with probability 0.746 (for $q = 2$, higher for bigger q). It is not hard to see that the minimal polynomial of M_B' divides that of M_B. There are therefore two possible situations:

1. Either M_b has only one invariant factor, and the rank of $\mathcal{S}$ is exactly $2n^2 - n$.
2. Or M_B has two invariant factors, x and $\chi_{M_B'}$, and the rank of $\mathcal{S}$ is $2n^2 - 2n + \operatorname{rank} \frac{\partial \mathbf{a}}{\partial y_1'}$. Lemma 6 then allows us to conclude that $\mathcal{S}$ has rank $2n^2 - n - 1$ or $2n^2 - n - 2$ with high probability.

4.3 The Affine Case and/or Homogeneous case

As pointed out in section 2, the affine case may be solved easily if the linear homogeneous case can be solved. Unfortunately, the two previous methods cannot be applied in the homogeneous case. It is therefore natural to try to "de-homogenize" the problem as was done for the cubic IP1S problem in section 3.2. This requires *one* known relation on S_ℓ: if we know $y_0 = S_\ell \cdot x_0$, then we define $\mathbf{a}'(\mathbf{x}) = \mathbf{a}(\mathbf{x} + y_0)$ and $\mathbf{b}'(\mathbf{x}) = \mathbf{b}(\mathbf{x} + x_0)$, and we have $\mathbf{b}' = T \circ \mathbf{a}' \circ S_\ell$. It follows that:

$$\mathbf{a}'^{(2)} = \mathbf{a}^{(2)} \qquad\qquad \mathbf{a}'^{(1)} = \frac{\partial \mathbf{a}}{\partial y_0}$$

A first consequence is that the homogeneous linear component of the de-homogenized version of the instance is never invertible in even characteristic. For this reason, the heuristic of section 4.1 is bound to fail. This again highlight the interest of the more robust algorithm of section 4.2. However, its rank analysis should be revised, because one of the assumption (the invertibility of the linear part) is no longer true. There is also a further difficulty. The algorithm of section 4.2 requires a boostrapping relation on $\mathcal{S}_\ell$. Using $y_0 = S_\ell \cdot x_0$ is not possible, as we would have $\mathbf{b}^{(1)} = \frac{\partial \mathbf{b}}{\partial x_0}$, and the two equations of (11) would in fact be the same. We know that $\mathbf{a}'(0) \neq 0$ yields a relation on T_ℓ, but this relation can only be transferred to S_ℓ if $\mathbf{a}'(0)$ belongs to the range of $\mathbf{a}'^{(1)}$. Since $\mathbf{a}'^{(1)}$ is not invertible in even characteristic, the probability that one relation on $\mathcal{S}_\ell$ is sufficient to solve the problem is at most $1/q$. Therefore, in even characteristic, there is a pretty non-negligible chance that *two* independent relations on S_ℓ are in fact needed. The problem is then to acquire the initial relation(s) on S_ℓ. Several options are possible:

1. Guess the image of S_ℓ on two chosen point, and run the algorithm of section 4.2. This method has complexity $\mathcal{O}\left(n^6 \cdot q^{2n}\right)$. If it succeeds, then the guesses were right.
2. Choose a point x_0 such that $\operatorname{rank} \frac{\partial B}{\partial x_0} = n-2$. Then, guess $y_0 = S \cdot x_0$. Because $\ker \frac{\partial \mathbf{a}}{\partial y_0} = S\left(\ker \frac{\partial \mathbf{b}}{\partial x_0}\right)$ is of dimension two, we find another relation on S. However, this yields a complex situation, where the two matrix equations of (11) are not statistically independent, and a dedicated rank analysis is required.
3. Find the non-trivial zeroes of $\mathbf{a}$ and $\mathbf{b}$. If there are k of them, then there are $\binom{k}{2}$ ways to match them to obtain two relations on S. The algorithm of section 4.2 has to be run for each combination.

The last method seems the most promising. It is known [24] that a random system of n polynomials in n variables has exactly $s \geq 0$ common zeroes with probability $1/(e \cdot s!)$. Finding the zeroes of $\mathbf{a}$ and $\mathbf{b}$ thus permits to find at least one relation on S_ℓ with probability 0.632, and two with probability 0.264. Thus, one fourth of the random instances can be solved in polynomial time *after the zeroes have been found.*

Depending on the value of q, solving quadratic equations will be faster with Gröbner bases (big q), or via exhaustive search (small q). For $q = 2$, a GPU-based implementation of exhaustive search (to appear) solves 48 equations in 48 variables in 30 minutes on a \$500 GPU (a Nvidia GT280). Thus the challenge with $n = 64$ can theoretically be broken with probability $1/4$ in less than 4 GPU-year with inexpensive hardware.

If only one relation is found, then all hope is not lost, because there are ways to compute a new relation from the first one with a certain probability, for instance by looking at the kernel of the differential. This naturally brings us to the study of an algorithm presented twelve years ago, that relied heavily on this idea.

4.4 About a Previous Algorithm

The algorithm we give for the hardest case has a complexity of order q^n. In the extended version of [41], an algorithm is given which supposedly solves this hardest case in time and memory $\mathcal{O}\left(q^{n/2}\right)$. This algorithm has never been implemented nor tested, and its practical impact is therefore unknown. It is highly heuristic, and its precise complexity and success probability are unknown. In this section, we point out several shortcomings in its description and analysis which convinced us that further analysis is required to fully understand its complexity and its behavior. More specifically, we will prove the following facts:

1. This algorithm fails systematically over $\mathbb{F}_q$ with $q = 2^m$ if $m > 1$.
2. One of the suggested "boosting functions" fails systematically
3. The non-failing one succeeds with probability 0.144 (instead of $1/2$ as claimed originally).
4. The complexity of the whole procedure is of order $\Omega\left(n^4 \cdot q^{n/2}\right)$.

We believe that this is enough to justify that this algorithm deserves a further analysis. Its study is non-trivial and makes an extensive use of lemma 6. Besides being an interesting problem that extends beyond the scope of this paper, the complete study of this algorithm raises many questions about the IP problem. It is yet completely unclear whether there exist a birthday-based algorithm with a complexity of order $\mathcal{O}\left(q^{n/2}\right)$. In the annex C, we justify the four points enumerated above.

5 Experimental Results

We report on experimental results obtained with our algorithms.

Cubic IP1S instances.

- $u = 1$, $n = 16$, $q = 2$: broken in 2 CPU-month (section 3.2).
- $u = 1$, $n = 6$, $q = 16$: Directly solving the first differential equation of section 3.2 for S and S^{-1} can be done in 490s using 259Mbyte using the F_4 algorithm.

Quadratic IP1S instances.

- $u = 2$, $n = 32$, $q = 2$. Broken instantly (section 3.1).
- $u = 2$, $n = 6$, $q = 16$. Idem.

Quadratic IP instances.

- $u = n = 64$ avec $q = 2$, linear inhomogeneous. Broken in negligible time (section 4.2)
- $u = n = 64$ avec $q = 2$, linear homogeneous. Broken in 4 GPU-year with probability 0.264 (section 4.3).
- $u = n = 8$, $q = 2^8$, homogeneous. Broken by [20] in time 5211s and space 2Gbyte.

6 Conclusion

In this paper, we present algorithms for the IP problem that solve the IP problem with one secret for random quadratic equations and one cubic equation and with two secrets. Moreover, we explain the complexity, success probability and give sufficient conditions so that the algorithms work. The basic idea of our algorithm is simple and some of them have been proposed for example on the IP with one secret but never formally analyzed. We try to find linear relations on the secrets and then apply Gröbner basis algorithm to linearize the system. In order to find linear relations we use the idea of Faugère and Perret and the differential of the systems. Finally, we highlight the following open problem that we face: is there a rigorous birthday paradox algorithm for the IP problem ?

References

1. Baena, J., Clough, C., Ding, J.: Square-vinegar signature scheme. In: PQCrypto '08: Proceedings of the 2nd International Workshop on Post-Quantum Cryptography, Berlin, Heidelberg, Springer-Verlag (2008) 17–30
2. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.Y.: Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In: MEGA'05. (2005) Eighth International Symposium on Effective Methods in Algebraic Geometry, Porto Conte, Alghero, Sardinia (Italy), May 27th – June 1st.
3. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. PhD thesis, Université de Paris VI (2004)
4. Bardet, M., Faugère, J.C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proc. International Conference on Polynomial System Solving (ICPSS). (2004) 71–75
5. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.Y.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In: Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry. (2005)
6. Bernstein, D.S.: Matrix mathematics. Theory, facts, and formulas. 2nd expanded ed. Princeton, NJ: Princeton University Press. xxxix, 1139 p. (2009)
7. Billet, O., Gilbert, H.: A traceable block cipher. In Laih, C.S., ed.: ASIACRYPT. Volume 2894 of Lecture Notes in Computer Science., Springer (2003) 331–346
8. Bosma, W., Cannon, J.J., Playoust, C.: The Magma Algebra System I: The User Language. J. Symb. Comput. **24**(3/4) (1997) 235–265
9. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD thesis, University of Innsbruck (1965)
10. Chevalley, C.: Démonstration d'une hypothèse de M. Artin. Abh. Math. Semin. Hamb. Univ. **11** (1935) 73–75
11. Clough, C., Baena, J., Ding, J., Yang, B.Y., Chen, M.S.: Square, a new multivariate encryption scheme. In Fischlin, M., ed.: CT-RSA. Volume 5473 of Lecture Notes in Computer Science., Springer (2009) 252–264
12. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties and Algorithms. Springer (2005)
13. dit Vehel, F.L., Perret, L.: Polynomial Equivalence Problems and Applications to Multivariate Cryptosystems. In Johansson, T., Maitra, S., eds.: INDOCRYPT. Volume 2904 of Lecture Notes in Computer Science., Springer (2003) 235–251
14. Dubois, V., Granboulan, L., Stern, J.: An efficient provable distinguisher for hfe. In Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., eds.: ICALP (2). Volume 4052 of Lecture Notes in Computer Science., Springer (2006) 156–167
15. Faugère, J.C., Gianni, P., Lazard, D., Mora, T.: Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. Journal of Symbolic Computation **16**(4) (1993) 329–344
16. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139**(1-3) (June 1999) 61–88

17. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139** (June 1999) 61–88

18. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In Mora, T., ed.: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation ISSAC, ACM Press (July 2002) 75–83 isbn: 1-58113-484-3.

19. Faugère, J.C., Joux, A.: Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases. In Boneh, D., ed.: Advances in Cryptology - CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 44–60

20. Faugère, J.C., Perret, L.: Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects. In Vaudenay, S., ed.: EUROCRYPT. Volume 4004 of Lecture Notes in Computer Science., Springer (2006) 30–47

21. Fortin, S.: The graph isomorphism problem. Technical report (1996)

22. Fouque, P.A., Macario-Rat, G., Stern, J.: Key Recovery on Hidden Monomial Multivariate Schemes. In Smart, N.P., ed.: EUROCRYPT. Volume 4965 of Lecture Notes in Computer Science., Springer (2008) 19–30

23. Fulman, J.: Random matrix theory over finite fields. Bull. Amer. Math. Soc. (N.S **39** 51–85

24. Fusco, G., Bach, E.: Phase transition of multivariate polynomial systems. Mathematical. Structures in Comp. Sci. **19**(1) (2009) 9–23

25. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of *NP*-Completeness. Freeman, New-York (1979)

26. Geiselmann, W., Meier, W., Steinwandt, R.: An Attack on the Isomorphisms of Polynomials Problem with One Secret. Int. J. Inf. Sec. **2**(1) (2003) 59–64

27. Geiselmann, W., Steinwandt, R., Beth, T.: Attacking the Affine Parts of SFLASH. In Honary, B., ed.: IMA Int. Conf. Volume 2260 of Lecture Notes in Computer Science., Springer (2001) 355–359

28. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. J. ACM **38**(3) (1991) 691–729

29. Granboulan, L., Joux, A., Stern, J.: Inverting HFE Is Quasipolynomial. In Dwork, C., ed.: CRYPTO. Volume 4117 of Lecture Notes in Computer Science., Springer (2006) 345–356

30. Koblitz, N.: Algebraic Aspects of Cryptography. Volume 3 of Algorithms and Computation in Mathematics. Springer-Verlag (1998)

31. Lazard, D.: Gröbner-bases, gaussian elimination and resolution of systems of algebraic equations. In van Hulzen, J.A., ed.: EUROCAL. Volume 162 of Lecture Notes in Computer Science., Springer (1983) 146–156

32. Lidl, R., Niederreiter, H.: Finite fields. Cambridge University Press, New York, NY, USA (1997)

33. MacWilliams, J.: Orthogonal matrices over finite fields. The American Mathematical Monthly **76**(2) (1969) 152–164

34. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Advances in Cryptology – EUROCRYPT 1988. Volume 330 of LNCS., Springer–Verlag (1988) 419–453

35. Naccache, D., ed.: Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001)

36. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: EUROCRYPT. (1996) 33–48 Etended version available on `http://www.minrank.org/hfe.pdf`.

37. Patarin, J.: Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In: EUROCRYPT. (1996) 33–48

38. Patarin, J.: The Oil and Vinegar signature scheme. presented at the Dagstuhl Workshop on Cryptography (1997)

39. Patarin, J., Courtois, N., Goubin, L.: Flash, a fast multivariate signature algorithm. [35] 298–307

40. Patarin, J., Courtois, N., Goubin, L.: QUARTZ, 128-Bit Long Digital Signatures. [35] 282–297

41. Patarin, J., Goubin, L., Courtois, N.: Improved Algorithms for Isomorphisms of Polynomials. In: EUROCRYPT. (1998) 184–200

42. Perret, L.: A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem. In Cramer, R., ed.: EUROCRYPT. Volume 3494 of Lecture Notes in Computer Science., Springer (2005) 354–370

43. Stong, R.: Some asymptotic results on finite vector spaces. Adv. Appl. Math. **9**(2) (1988) 167–199

44. Warning, E.: Bemerkung zur vorstehenden Arbeit von Herrn Chevalley. Abh. Math. Semin. Hamb. Univ. **11** (1935) 76–83

45. Wolf, C., Preneel, B.: Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations. Cryptology ePrint Archive, Report 2005/077 (2005) `http://eprint.iacr.org/`.

A Some Proofs

A.1 Proof of lemma 1

Proof.

$$\begin{aligned}
\mathbf{b}(x) &= T_c + T_\ell \cdot \mathbf{a}(S_\ell \cdot x + S_c) \\
&= T_c + T_\ell \cdot \Big(\mathrm{D}\mathbf{a}(S_\ell \cdot x, S_c) + \mathbf{a}(S_\ell \cdot x) + \mathbf{a}(S_c) - \mathbf{a}(0) \Big) \\
&= \Big[T_c + T_\ell \cdot \big(\mathbf{a}\left(S_c\right) - \mathbf{a}(0) \big) \Big] + \left(T_\ell \circ \frac{\partial \mathbf{a}}{\partial S_c} \circ S_\ell \right)(x) + \left(T_\ell \circ \mathbf{a} \circ S_\ell \right)(x)
\end{aligned}$$

The first statement follows from the application of [20, lemma 4] to the last equality. The second statement is a direct consequence of the first one. □

A.2 Proof of theorem 1

Proof. A single matrix equation $X \cdot B - A \cdot Y = 0$ yields structured equations between the $X_{i,j}$ and the $Y_{i,j}$. Let us denote by M_i the i-th row of the matrix M. The equation $X \cdot B - A \cdot Y = 0$ translates to:

$$\left(\begin{array}{ccc|ccc}
 & & & A_1 & & \\
{}^tB & & & & \ddots & \\
 & & & & & A_1 \\
\hline
 & & & A_2 & & \\
 & {}^tB & & & \ddots & \\
 & & & & & A_2 \\
\hline
 & \ddots & & & \vdots & \\
\hline
 & & & A_n & & \\
 & & {}^tB & & \ddots & \\
 & & & & & A_n
\end{array}\right) \times \left(\begin{array}{c}
X_1 \\ \hline \vdots \\ \hline X_n \\ \hline {}^tY_1 \\ \hline \vdots \\ \hline {}^tY_n
\end{array}\right) = 0$$

The equations defining $\mathcal{S}$ thus have a nice block structure, but to make it simpler, we define the following $n \times n^2$ block-matrices:

$$P_j = \begin{pmatrix} A_1 \cdot e_j & & \\ & \ddots & \\ & & A_1 \cdot e_j \end{pmatrix} \qquad Q_j = \begin{pmatrix} A_2 \cdot e_j & & \\ & \ddots & \\ & & A_2 \cdot e_j \end{pmatrix}$$

And then we see that:

$$\mathcal{S} : \quad \left(\begin{array}{cccc|c}
{}^tB_1 & & & & P_1 \\
{}^tB_2 & & & & Q_1 \\
\hline
 & {}^tB_1 & & & P_2 \\
 & {}^tB_2 & & & Q_2 \\
\hline
 & & \ddots & & \vdots \\
\hline
 & & & {}^tB_1 & P_n \\
 & & & {}^tB_2 & Q_n
\end{array}\right) \times \left(\begin{array}{c}
X_1 \\ \hline \vdots \\ \hline X_n \\ \hline {}^tY_1 \\ \hline \vdots \\ \hline {}^tY_n
\end{array}\right) = 0$$

We now proceed through the echelonization of $\mathcal{S}$. Let us define the following partial echelonization matrix:

$$\varDelta = \left(\begin{array}{c|c|c|c|c|c|c}
{}^tB_1^{-1} & & & & & & \\
M_B & 1_n & & & & & \\
\hline
 & & {}^tB_1^{-1} & & & & \\
 & & M_B & 1_n & & & \\
\hline
 & & & & \ddots & & \\
\hline
 & & & & & {}^tB_1^{-1} & \\
 & & & & & M_B & 1_n
\end{array}\right)$$

$\varDelta$ is invertible, since it is block-lower-triangular, and all the diagonal blocks are invertible. Therefore $\mathcal{S}$ has the same rank as:

$$\mathcal{S}' = \varDelta \times \mathcal{S} = \left(\begin{array}{ccc|c} 1_n & & & {}^tB_1^{-1}P_1 \\ 0_n & & & Q_1 + M_BP_1 \\ \hline & 1_n & & {}^tB_1^{-1}P_2 \\ & 0_n & & Q_2 + M_BP_2 \\ \hline & \ddots & & \vdots \\ \hline & & 1_n & {}^tB_1^{-1}P_n \\ & & 0_n & Q_n + M_BP_n \end{array}\right)$$

Reordering the rows yields:

$$\mathcal{S}'' = \left(\begin{array}{cccc|c} 1_n & & & & {}^tB_1^{-1}P_1 \\ & 1_n & & & {}^tB_1^{-1}P_2 \\ & & \ddots & & \vdots \\ & & & 1_n & {}^tB_1^{-1}P_n \\ \hline & & & & Q_1 + M_BP_1 \\ & & & & Q_2 + M_BP_2 \\ & & & & \vdots \\ & & & & Q_n + M_BP_n \end{array}\right)$$

Here, we give some notations. Let A and M be two matrices over $\mathbb{F}_q$ of dimension $n \times n$. We denote by $A \otimes M$ the $n^2 \times n^2$ matrix with the following block-structure: the $n \times n$ block starting at row $(i-1)n+1$ and column $(j-1)n+1$ is $M_{i,j} \cdot A$. It is not difficult to see that $\operatorname{rank} A \otimes M = (\operatorname{rank} A)(\operatorname{rank} M)$.

And we are therefore left with the task of estimating the rank of the lower-right block, which we denote by $\mathcal{T}$ (it has dimension $n^2 \times n^2$). Reordering its rows (more precisely, swapping row $(i-1)n+j$ and row $(j-1)n+i$) shows a different, more complicated, block-structure:

$$\mathcal{T}' = A_1 \otimes M_B + A_2 \otimes 1_n$$

Since A_1 is invertible as well, we obtain:

$$\mathcal{T}' = (A_1 \otimes 1_n) \cdot (1_n \otimes M_B + M_A \otimes 1_n)$$

The left member of this product is obviously invertible, therefore, the rank of $\mathcal{T}$ is the rank of $\mathcal{T}'' = 1_n \otimes M_B + M_A \otimes 1_n$. Let us define U, a $n \times n$ matrix over $\mathbb{F}_q[x]$:

$$U = M_B - x \cdot 1_n$$

Then the $n \times n$ matrix $U_{i,j}(M_A)$ is precisely the $n \times n$ block of indices (i,j) of $\mathcal{T}''$. This allows us to work on U and lift the result back to $\mathcal{T}''$. The determinant of U, which is a polynomial of degree n, is precisely the characteristic polynomial of M_B (which we denote χ_{M_B}). The *Smith Normal Form* of U is:

$$U' = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & p_1 & & \\ & & & & \ddots & \\ & & & & & p_s \end{pmatrix}$$

And there exist two unimodular matrices (*i.e.*, with determinant one) V and V' such that $U = V \cdot U' \cdot V'$. The monic polynomials $p_1, \ldots, p_n \in \mathbb{F}_q[x]$ are such that p_i divides p_{i+1}. In fact, $U'_{i,j}(B)$ is simply the block of indices (i,j) of:

$$\mathcal{T}''' = (1_n \otimes V^{-1}) \cdot \mathcal{T}'' \cdot (1_n \otimes V'^{-1}).$$

This means that $\mathcal{T}$ and T''' have the same rank. Because $\mathcal{T}'''$ is block-diagonal, its rank is the sum of the rank of the diagonal elements, namely $U'_{ii}(M_A)$. This concludes the proof. □

A.3 Proof of lemma 4

Proof. We consider the coefficient of the monomial $m = y_1 y_i x_j$ in the expression

$$\varDelta = \varDelta_a - \varDelta_b = \mathrm{D}\mathbf{a}(S_0(S+I_n)\mathbf{x},\mathbf{y}) - \mathrm{D}\mathbf{b}(\mathbf{x},(U+I_n)S_0^{-1}\mathbf{y}).$$

Since the monomial m is linear in x_j it is clear that the corresponding coefficient in $\varDelta_a = \mathrm{D}\mathbf{a}(S_0(S+I_n)\mathbf{x},\mathbf{y})$ is also linear in the variables $s_{i,j}$; moreover this coefficient is non zero. We have now to consider the coefficient of m in $\varDelta_b$. Since $\mathrm{D}\mathbf{b}(\mathbf{x},\mathbf{y})$ is the differential of an homogenous polynomial of degree 3 we can always write:

$$\mathrm{D}\mathbf{b}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \ell_{i,j}(y_1,\ldots,y_n)\, x_i x_j + \sum_{i=1}^{n} q_i(y_1,\ldots,y_n) x_i \tag{12}$$

where $\ell_{i,j}$ (resp. q_i) is a polynomial of degree 1 (resp. 2). Consequently, the coefficient of m in $\mathrm{D}\mathbf{b}$ is also the coefficient of $y_1 y_i$ in $q_j((U+I_n)S_0^{-1}\mathbf{y})$. That is to say, in $q_j(\mathbf{y})$ we have now to replace $\mathbf{y} = (y_1,\ldots,y_n)$ by $((U+I_n)S_0^{-1}\mathbf{y})$. Thus, modulo the equations of the ideal $\mathcal{J}'$ and (6), we can write the product $((U+I_n)S_0^{-1}\mathbf{y})$ as

$$= \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ u_{2,1} & \cdots & \cdots & u_{2,n} \\ \vdots & \cdots & \cdots & \vdots \\ u_{n,1} & \cdots & \cdots & u_{n,n} \end{pmatrix} \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$

$$= \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} \begin{pmatrix} * & * & * & * \\ (*u_{2,1} + \cdots + *u_{2,n}) & \cdots & \cdots & (*u_{2,1} + \cdots + *u_{2,n}) \\ \vdots & \cdots & \cdots & \vdots \\ (*u_{n,1} + \cdots + *u_{n,n}) & \cdots & \cdots & (*u_{2,1} + \cdots + *u_{n,n}) \end{pmatrix}$$

$$= \begin{pmatrix} *y_1 + (*u_{2,1} + \cdots + *u_{2,n})y_2 + \cdots + (*u_{n,1} + \cdots + *u_{n,n})y_n \\ *y_1 + (*u_{2,1} + \cdots + *u_{2,n})y_2 + \cdots + (*u_{n,1} + \cdots + *u_{n,n})y_n \\ \vdots \\ *y_1 + (*u_{2,1} + \cdots + *u_{2,n})y_2 + \cdots + (*u_{n,1} + \cdots + *u_{n,n})y_n \end{pmatrix}$$

Hence the coefficient of $y_1 y_i$ in $q_j((U+I_n)S_0^{-1}\mathbf{y})$ is linear in the variables $u_{k,l}$ when $i \neq 1$ and the coefficient of y_1^2 is a constant. □

B Probabilities For Random Matrices

If q is odd, then the probability that a random symmetric matrix has a given rank is given by the following result.

Lemma 7 ([33], theorem 2). *Let $N(n,r)$ denote the number of symmetric matrices of size $n \times n$ over $\mathbb{F}_q$ and of rank r.*

$$N(n,2s) = \prod_{i=1}^{s} \frac{q^{2i}}{q^{2i}-1} \cdot \prod_{i=1}^{2s-1} \left(q^{n-i}-1\right)$$

$$N(n,2s+1) = \prod_{i=1}^{s} \frac{q^{2i}}{q^{2i}-1} \cdot \prod_{i=1}^{2s} \left(q^{n-i}-1\right)$$

If q is even, then the probability that a random symmetric matrix with zeros on the diagonal, has a given rank is given by this other result.

Lemma 8 **([33], theorem 3).** *Let $N_0(n, r)$ denote the number of symmetric matrices of size $n \times n$ over $\mathbb{F}_q$ with zeros on the diagonal and of rank r.*

$$N_0(n, 2s) = \prod_{i=1}^{s} \frac{q^{2i-2}}{q^{2i}-1} \cdot \prod_{i=1}^{2s-1} \left(q^{n-i} - 1\right)$$
$$N_0(n, 2s+1) = 0$$

Lemma 9 **([23], theorem 1).** *Let $c(n, q)$ be the proportion of cyclic $n \times n$ matrices (i.e., matrices for which the minimal polynomial is of degree n). We have:*

$$\frac{1}{q^2(q+1)} < 1 - c(n, q) < \frac{1}{(q^2-1)(q-1)}$$

And asymptotically, we have:

$$\lim_{n \to \infty} c(n, q) = \frac{q^5 - 1}{q(q-1)(q^2-1)} \cdot \prod_{i=1}^{\infty} \left(1 - \frac{1}{q^i}\right)$$

C A Birthday Paradox Algorithm?

Complete Failure over $\mathbb{F}_{2^m}$, $m > 1$. This algorithm makes a critical use of so-called "boosting functions". A boosting function is in fact a pair of functions $F_{\mathbf{a}}$ and $F_{\mathbf{b}}$ such that $S_\ell \cdot \mathbf{x} = \mathbf{y}$ implies $S_\ell \cdot F_{\mathbf{b}}(\mathbf{x}) = F_{\mathbf{a}}(\mathbf{y})$ with non-negligible probability. The evaluation of $F_{\mathbf{a}}$ and $F_{\mathbf{b}}$ must be efficient (typically, polynomial in the parameters), and the functions are not assumed to be deterministic. It is expected that $F_{\mathbf{b}}(\mathbf{x})$ is different from both zero and $\mathbf{x}$ with a noticeable probability (otherwise, they do not help much).

The starting point over which boosting functions are constructed in [41] is the observation that for a given vector $\mathbf{z}$, the sets $E_{\mathbf{a}} = \{\mathbf{y} | \mathbf{a}(S_\ell \cdot \mathbf{z} + \mathbf{y}) = \mathbf{a}(\mathbf{y})\}$ and $E_{\mathbf{b}} = \{\mathbf{x} | \mathbf{b}(\mathbf{z} + \mathbf{x}) = \mathbf{b}(\mathbf{x})\}$ are in correspondence via S_ℓ: $E_{\mathbf{a}} = S_\ell(E_{\mathbf{b}})$. This equality between sets can be collapsed to a single vector equality by summing over all the elements:

$$S_\ell \cdot \left(\sum_{\mathbf{y} \in E_{\mathbf{b}}} \mathbf{y} \right) = \sum_{\mathbf{x} \in E_{\mathbf{a}}} \mathbf{x}$$

.

This would naturally lead to the definition of a boosting function:

$$F_{\mathbf{a}}(\mathbf{z}) = \sum_{\mathbf{a}(\mathbf{y}+\mathbf{z})=\mathbf{a}(\mathbf{y})} \mathbf{y} \qquad\qquad F_{\mathbf{b}}(\mathbf{z}) = \sum_{\mathbf{b}(\mathbf{x}+\mathbf{z})=\mathbf{b}(\mathbf{x})} \mathbf{x}$$

Unfortunately, as the authors of [41] pointed out, over fields of characteristic two, we have that if $\mathbf{x} \in E_{\mathbf{b}}$, then so does $\mathbf{x} + \mathbf{z}$, and the elements of $E_{\mathbf{b}}$ would always sum to a vector collinear to $\mathbf{z}$ (a similar result applies to $E_{\mathbf{a}}$). In any case, no new information on S_ℓ can be obtained this way. To overcome this issue, they suggested to sum over the elements of $E_{\mathbf{b}}$, but excluding $\mathbf{x} + \mathbf{z}$ if $\mathbf{x}$ enters the sum. More precisely, they defined:

$$F_{\mathbf{a}}(\mathbf{z}) = \sum_{\substack{\mathbf{a}(\mathbf{y}+\mathbf{z})=\mathbf{a}(\mathbf{y}) \\ \text{one only} \in \{\mathbf{y}, \mathbf{y}+\mathbf{z}\}}} \mathbf{y} \qquad\qquad F_{\mathbf{b}}(\mathbf{z}) = \sum_{\substack{\mathbf{b}(\mathbf{x}+\mathbf{z})=\mathbf{b}(\mathbf{x}) \\ \text{one only} \in \{\mathbf{x}, \mathbf{x}+\mathbf{z}\}}} \mathbf{x}$$

In other terms, they suggested using the algorithm shown in figure 2 to compute the sum. In other terms, $F_{\mathbf{a}}(\mathbf{z}) = \mathbf{PartialSum}(E_{\mathbf{a}}, \mathbf{z})$ and $F_{\mathbf{b}}(\mathbf{z}) = \mathbf{PartialSum}(E_{\mathbf{b}}, \mathbf{z})$. Unfortunately again, this way of doing things does not solve the problem, as we now establish. For this purpose, we first prove a general result on $\mathbf{PartialSum}$.

Lemma 10. *Let V be a vector space such that $\mathbf{z} \in V$, and c be a vector in $(\mathbb{F}_q)^n$. We denote by $c + V$ the affine space formed by the vectors $c + x$, for all $x \in V$.*

i) If $q > 2$ or $\dim V > 1$, then $\mathbf{PartialSum}(c + V, \mathbf{z}) \in \mathrm{Span}(\mathbf{z})$.
ii) If $q = 2$ and $\dim V \leq 1$, then $\mathbf{PartialSum}(c + V, \mathbf{z})$ *returns either c or $c + \mathbf{z}$*

Fig. 2 : $\mathbf{PartialSum}(V, \mathbf{z})$

```
1: $Ban \leftarrow \emptyset$
2: $sum \leftarrow 0$
3: while $V - Ban \neq \emptyset$ do
4:     let $\mathbf{x}$ be a random element of $(V - Ban)$
5:     $Ban \leftarrow Ban \cup \{\mathbf{x}, \mathbf{x} + \mathbf{z}\}$
6:     $sum \leftarrow sum + \mathbf{x}$
7: end while
8: return $sum$
```

Proof. Let us denote by k the dimension of V. As a consequence, we have $\left|V\right| = q^k$. The vectors that are added to sum on line 6 are of the form : $c + v$, with $v \in V$, and the "while" loop on line 3 is iterated $q^k/2$ times.

If $q = 2$ and $k = 1$, the loop is iterated only once, and since $V = \{0, \mathbf{z}\}$, either c or $c + \mathbf{z}$ will be selected, which proves the second part of our statement.

The number of iterations of the loop is even as soon as $q > 2$ or $\dim V > 1$. If either one of these conditions is true, then c being added an even number of times to sum cancels out, and we have that:

$$\mathbf{PartialSum}(c + V, \mathbf{z}) = \mathbf{PartialSum}(V, \mathbf{z})$$

Next, it is known that $\mathbf{z} \neq 0$ belongs to V. Therefore, we can extend $\mathbf{z}$ to a basis of V. More precisely, let us denote by $U = \mathrm{Span}\,(b_1, b_2, \ldots, b_{k-1})$ a $(k-1)$-dimensional vector space such that $V = \mathrm{Span}(\mathbf{z}) \bigoplus U$. Let us denote by ρ the projection from V to U (*i.e.*,, the mapping that keeps only the last $k-1$ coordinates). To establish the first part of our statement, we need to show that $\rho(sum) = 0$ at the end of the loop.

Now, there are exactly q vectors in V that are sent to the same element α of U by ρ. During the loop, $q/2$ of these vectors will be selected (the other half being discarded).

- If $q = 2$, exactly one vector the projection of which is α will be selected, for all $\alpha \in U$. Therefore,

$$\rho(sum) = \sum_{\alpha \in U} \alpha = 0$$

- If $q > 2$, because $k > 1$, then an even number of vectors with the same projection will be selected in the loop, and therefore their projetions sum up to zero. □

Corollary 1. *Unless $q = 2$ and the rank of $\frac{\partial \mathbf{b}}{\partial \mathbf{z}}$ is $n - 1$, $F_{\mathbf{a}}(\mathbf{z})$ is collinear to* $\mathbf{z}$*.*

Proof. The set $E_{\mathbf{b}}$ is in fact an affine space: it is easy to see that $\mathbf{x} \in E_{\mathbf{b}}$ is equivalent to $\mathbf{b}(\mathbf{z}) + \mathrm{D}_{\mathbf{z}} b \cdot \mathbf{x} = 0$. This equation yields u affine relations, the solution of which can be written:

$$E_{\mathbf{b}} = c + \ker \mathrm{D}_{\mathbf{z}} b$$

where c denotes a particular solution. It must be noted that $\mathbf{z} \neq 0$ belongs to $\ker \mathrm{D}_{\mathbf{z}} b$. Then, lemma 10 concludes the proof. □

We have shown that as soon as $q > 2$, this particular boosting function cannot, by any means, provide additional knowledge on S_ℓ, and when using it, the while algorithm is bound to fail. On the other hand, it was also bound to fail when $\mathbf{a}$ and $\mathbf{b}$ are injective mappings (since in this case, $E_{\mathbf{a}} = E_{\mathbf{b}} = \emptyset$). Note that the same reasoning about the size of the field can be extended to any boosting function making use of $\mathbf{PartialSum}$.

Systematic Failure of a Boosting Function. An other boosting function is defined in [41], with the aim of overcoming this last problem:

$$G_{\mathbf{a}}(\mathbf{z}) = \sum_{\substack{\mathbf{a}(\mathbf{y}+\mathbf{z})=\mathbf{a}(\mathbf{y})+\mathbf{a}(\mathbf{z}) \\ \text{one only} \in \{\mathbf{y}, \mathbf{y}+\mathbf{z}\}}} \mathbf{y} \qquad G_{\mathbf{b}}(\mathbf{z}) = \sum_{\substack{\mathbf{b}(\mathbf{x}+\mathbf{z})=\mathbf{b}(\mathbf{x})+\mathbf{b}(\mathbf{z}) \\ \text{one only} \in \{\mathbf{x}, \mathbf{x}+\mathbf{z}\}}} \mathbf{x}$$

Or, using our terminology:

$$G_{\mathbf{a}}(\mathbf{z}) = \mathbf{PartialSum}\left(\ker \frac{\partial \mathbf{a}}{\partial \mathbf{z}}, \mathbf{z}\right) \qquad G_{\mathbf{b}}(\mathbf{z}) = \mathbf{PartialSum}\left(\ker \frac{\partial \mathbf{b}}{\partial \mathbf{z}}, \mathbf{z}\right)$$

Unfortunately, G cannot be used as a boosting function at all, even if $q = 2$.

Lemma 11. *$G_{\mathbf{b}}(\mathbf{z})$ is collinear to $\mathbf{z}$.*

Proof. Here, we have $E_{\mathbf{b}} = \ker \mathrm{D}_{\mathbf{z}}\mathbf{b}$. By lemma 10 (and in either of the considered cases), we obtain the result. □

Success Probability of the Boosting Function. So far, we have shown that the algorithm proposed in [41] does not work at all over $\mathbb{F}_q$ with $q = 2^k > 2$. Therefore, it cannot be used to break some of the challenges proposed by Patarin. On the positive side, it was shown that boosting functions do exist. It is even possible to study them rigorously.

Lemma 12. *When $q = 2$, and assuming that $\mathbf{b}$ is chosen randomly amongst all quadratic maps, and that $\mathbf{z}$ is randomly chosen, the success probability of the boosting function is about 0.144 (and not always $1/2$ as incorrectly stated in [41]).*

Proof. If $q = 2$, the probability that $\dim\ker \frac{\partial \mathbf{b}}{\partial \mathbf{z}} = 1$, for a random $\mathbf{z}$, is $2\lambda(n)$ according to lemma 6.

If $q = 2$ and $\dim\ker \frac{\partial \mathbf{b}}{\partial \mathbf{z}} = 1$, then $E_{\mathbf{b}}$ will be empty if the affine equation $\mathbf{b}(\mathbf{z}) + \frac{\partial \mathbf{b}}{\partial \mathbf{z}} \cdot \mathbf{x} = 0$ has no solutions. It has solutions if and only if $\mathbf{b}(\mathbf{z})$ belongs to the range of $\frac{\partial \mathbf{b}}{\partial \mathbf{z}}$. Because $\frac{\partial \mathbf{b}}{\partial \mathbf{z}}$ is of rank $n - 1$, a random vector belongs to its range with probability $1/2$. Now, $\frac{\partial \mathbf{b}}{\partial \mathbf{z}}$ and $\mathbf{b}(\mathbf{z})$ are not statistically independent, but they are nevertheless empirically sufficiently independent for the observed probability to be $1/2$.

If it is not empty, then $E_{\mathbf{b}} = \{c, c + \mathbf{z}\}$, and sum will be chosen randomly amongst these two vectors. Since a random choice also takes place in $F_{\mathbf{a}}$, the probability that the boosting function provides a right relation for S is $1/2$, under these hypotheses.

So all in all, the success probability is exactly $\lambda(n)/2$. This quantity is a decreasing function of n that quickly converges to its limit (0.144). This is confirmed by experiments. □

It is not very hard to improve on those presented in [41]. More specifically, it is possible, based on the same ideas, to build a boosting function that succeeds 3 times more often that $F_{\mathbf{a}}$ (by simply returning a random vector from $\ker \frac{\partial \mathbf{a}}{\partial \mathbf{z}} - \{\mathbf{z}\}$).

Complexity of $\boldsymbol{\Omega\left(n^4 \cdot q^{n/2}\right)}$. Evaluating the boosting function $F_{\mathbf{a}}$ takes $\mathcal{O}\left(n^3\right)$ operations, as it requires solving a system of n linear equations in n unknowns. It is clear from the (vague) description of the algorithm in [41] that the boosting function $H_{\mathbf{a}}$ is evaluated at least $n \cdot q^{n/2}$ times.